LOGO Writer®

Reference Guide

Logo Computer Systems Inc.

LOGO Writer®

Reference Guide

© Logo Computer Systems Inc. 1988
All rights reserved.

No part of the documents contained herein may be reproduced, stored in retrieval systems or transmitted, in any form or by any means, photocopying, electronic, mechanical, recording or otherwise, without the prior approval in writing from Logo Computer Systems Inc.

Legal deposit, 2nd semester 1987
3rd revision, August 1988

ISBN 2-89371-004-2
ISBN 2-89271-005-0

The LCSI logo and LogoWriter are registered trademarks of Logo Computer Systems Inc.
Apple, the Apple logo, and ProDOS are registered trademarks and Apple *IIe*, Apple *IIc*, Apple IIGS and AppleWorks are trademarks of Apple Computer, Inc.
COMMODORE, COMMODORE 64, and the COMMODORE logo are registered trademarks of Commodore Electronics Limited and/or Commodore Business Machines Inc.
IBM is a registered trademark and IBM PC*jr* is a trademark of International Business Machines Corporation.
MS-DOS is a registered trademark of Microsoft Corporation.

Printed in Canada

Table of Contents

Preface

Section 1: Overview

Overview 1-9

I. Using the LogoWriter Scrapbook 1-11
Contents 1-11
A Scrapbook Page 1-12
Saving a Page 1-12
Protecting Your Pages 1-13
The Page and the Workspace 1-13
Saving Variables 1-13
Saving Events 1-13
Special Pages 1-14
Shapes 1-14
Help 1-14
Startup 1-15
Changing Scrapbook Disks 1-15
Copying Pages 1-15
Copying A Page Onto Another Disk 1-16
Copying A Shapes Page Onto Another Disk 1-16
Copying Pages With Two Disk Drives 1-17

II. LogoWriter Keys 1-19
General Purpose Keys 1-19
Help Keys 1-20
Graphics Keys 1-20
Word Processing Keys 1-21
Shapes Page Keys 1-22
Contents Keys 1-24

Section 2: LogoWriter Primitives

LogoWriter Primitives 2-1

Input Words 2-2
Infix Math Reporters 2-5
A 2-7
B 2-11
C 2-15
D 2-29
E 2-31
F 2-35
G 2-39
H 2-41
I 2-43
K 2-47
L 2-49
M 2-53
N 2-55
O 2-59
P 2-61
R 2-69
S 2-77
T 2-93
U 2-101
W 2-103
X 2-107
Y 2-109
ProDOS® Primitives 2-111
IBM DOS Primitives 2-121
System Primitives 2-129
System Primitives for Apple *II* Computers 2-129
System Primitives for IBM Computers 2-131

Section 3: LogoWriter Versions

LogoWriter Versions 3-1

I. LogoWriter 2.0 for Apple *II* Computers 3-3
LogoWriter Keys 3-4
Apple - Combination Keys 3-4

Apple LogoWriter Colors ____ 3-5
Game Paddle Inputs ____ 3-5
The ProDOS Features of LogoWriter 2.0 ____ 3-6
Files and Pages ____ 3-6
Disks and Volumes:
Setdisk Versus Setprefix ____ 3-7
Directories ____ 3-8
Pathnames ____ 3-11
ProDOS Hierarchy and
LogoWriter Commands ____ 3-15
Using the ProDOS Primitives ____ 3-16

II. LogoWriter 2.0 for IBM Computers ____ 3-19
LogoWriter Keys ____ 3-20
Function Keys ____ 3-20
Ctrl-Combination Keys ____ 3-20
Special Keys ____ 3-21
Game Port Inputs ____ 3-21
Game Paddles ____ 3-21
Joysticks ____ 3-22
LogoWriter Graphics Colors ____ 3-22
IBM Color Graphics Adapter ____ 3-22
Enhanced Graphics Adapter ____ 3-23
PS/2 Graphics ____ 3-24
The Vertical Bar (|) Delimiter
Character ____ 3-25
DOS Features: Subdirectory and File
Management ____ 3-26
Files and Pages ____ 3-26
Disks and Directories ____ 3-27
Summary ____ 3-30
A Special Note About IBM DOS ____ 3-31
Creating and Erasing Subdirectories ____ 3-31
Subdirectories and LogoWriter
Commands ____ 3-34

III. LogoWriter 1.1 ____ 3-37
About LogoWriter 1.1 ____ 3-37
Why Use LogoWriter 1.1? ____ 3-37
Last Page ____ 3-38
LogoWriter 1.1 for Apple *II*
Computers ____ 3-39
Memory Requirements ____ 3-39
LogoWriter DOS ____ 3-39
Converting Pages ____ 3-39

LogoWriter for the Apple *II+* 3-40
Making Copies 3-40
Exchanging Disks 3-40
Key Use 3-40
LogoWriter 1.1. for the IBM PC and IBM PC*jr* 3-43
Common Features 3-43
Making Sounds 3-43
IBM PC LogoWriter Colors 3-44
LogoWriter 1.1 for the IBM PC*jr* 3-46
Fn -Combination Keys 3-46
Ctrl-Combination Keys 3-47
PC*jr* LogoWriter Colors 3-47
LogoWriter 1.1 for the COMMODORE 64 Computer 3-49
COMMODORE - Combination Keys 3-49
COMMODORE 64 LogoWriter Colors 3-50
Game Paddle Inputs 3-51

Section 4: Appendices

Appendices 4-1

1. **LogoWriter Primitives and Special Words** 4-3

2. **Startup** 4-7
Startup as a Page Name 4-7
Startup as a Procedure Name 4-8
Startup Hints 4-8

3. **Tools** 4-9
What are Tools? 4-9
Advantages of Tools 4-9
Making a Page of Tools 4-10
Debugging Tools 4-10
Duplicate Procedure Names 4-11
Get Tools With Startup 4-11
Tools Pages on the Master Disk 4-12

4. **LogoWriter Messages** 4-13

5. **Glossary of LogoWriter and Computer Terms** 4-21

Index 4-29

Preface

The *LogoWriter ® Reference Guide* is a complete, in-depth source of information on all LogoWriter primitives, as well as a guide to some of the more technical aspects of LogoWriter.

The manual is divided into four sections:

Section 1: *Overview*
Provides general information on how to use the LogoWriter scrapbook, and what keys LogoWriter uses for its special functions.

Section 2: *LogoWriter Primitives*
Provides a description of each LogoWriter primitive including an example of its use. The primitives are in alphabetical order, with two extra chapters describing primitives which are not available in all versions of LogoWriter.

Section 3: *LogoWriter Versions*
Describes in detail the differences between the various versions of LogoWriter, and how LogoWriter uses the keyboard, screen, and other hardware on different computers.

Section 4: *Appendices*
Provides information on technical issues and the special features, Tools and Startup. It also includes a complete listing of the primitives by function and a glossary.

Section 1: Overview

This section is an overview of the fundamentals of LogoWriter.

Chapter I, *Using the LogoWriter Scrapbook*, describes the scrapbook, the pages of a scrapbook, and how to transfer pages from one scrapbook to another.

Chapter II, *LogoWriter Keys*, describes the function of every special key used by LogoWriter, whether for choosing or erasing pages, or for word processing or graphics.

1 Using the LogoWriter Scrapbook

A *scrapbook* is a set of LogoWriter pages on one disk. Each page in the scrapbook is a file on the disk. A disk may contain the LogoWriter program and a scrapbook, or a scrapbook only (without the LogoWriter program).

One scrapbook contains:

- The Contents page.
- The Shapes page.
- A number of pages containing graphics, text, and LogoWriter procedures. If you've just started using LogoWriter, these pages may include the Learn tutorial pages.

A scrapbook may contain:

- A Help page.
- A Startup page.

Pages can be easily copied from one scrapbook to another. Scrapbooks can be changed without restarting LogoWriter.

Note: In LogoWriter 2.0, a scrapbook disk can be divided into subdirectories, each of which functions like a scrapbook of its own. This feature is not discussed in this chapter. For more information on using subdirectories, refer to the chapter for your type of computer in the section *LogoWriter Versions*.

Contents

Contents is a menu page that lists the pages in the scrapbook. On the Contents page, you can choose a page or erase one or more pages.

To choose a page, use the **up** and **down arrow** keys to move the cursor onto the page name, and press Return Enter

To erase a page from the scrapbook, move the cursor onto the page name and press the **erase to end of line** keys (⌘-6 / F-6 / ⌥-6).

Warning: Erasing a page is permanent. An erased page cannot be recovered.

A Scrapbook Page

A page has two sides:

The *front side* of the page, where text is written and graphics are drawn.

The *flip side*, where procedures are defined.

The *command center* is where instructions are typed. It is not part of the page.

Saving a Page

To save a page in the scrapbook, either press Escape to go to Contents, or use **getpage** or **newpage** to get another page. The command **savepage** saves a page without leaving the page. In every case, the page must be named with **namepage** before it can be saved.

LogoWriter will save a page even if procedures on the flip side of that page contain bugs. If LogoWriter finds a bug when getting a page, a message is printed as a warning. The bug should be fixed as soon as possible, since the procedure with the bug *and any procedures following it* will not be in your workspace.

Protecting Your Pages

Once a page is created and is "finished," to the point that you don't want it to be changed, it can be protected against accidental changes. The command **lock** "write-protects" a page: the page can be looked at, but any changes are not saved. As a reminder, the word "locked" appears on the title line of a locked page.

The Page and the Workspace

Your workspace is the part of the computer's memory that is available for your use.

The size of a page is not the same as the size of your workspace. The text on the front of the page is not part of the workspace. All of the procedures on the flip side of the page are in the workspace. The workspace can also contain things which are not on the flip side of the page: variables, control-key events and tool procedures. Tool procedures are described in more detail in Appendix 3, *Tools*.

When flipping to the front of the page, there is sometimes a delay of a few seconds. That happens because changes or additions have been made to the procedures on the flip side, and LogoWriter is modifying the procedures in the workspace.

Saving Variables

Variables are not saved on a page. To be saved, the **make** or **name** instructions must be in procedures.

Saving Events

Control-key events (created with the **when** command) are not saved on a page. To be saved, the **when** instructions must be in procedures.

Special Pages

Shapes

To get the Shapes page, choose Shapes on the Contents page, or use the **shapes** command. All the shapes in a scrapbook are shown on the Shapes page. Shapes are created and changed on the flip side of the Shapes page. Press Escape to leave the Shapes page.

When you change scrapbook disks, the shapes in the computer's memory (the shapes that LogoWriter knows about) are not changed until you go to the Shapes page. In LogoWriter 2.0, there is a command (**getshapes**) that can change the shapes without going to the Shapes page.

Help

The Help page is intended to be a page of reminders for students. The information on the Help page can be easily modified, so teachers can create their own Help pages appropriate to a student's progress.

Any LogoWriter disk or scrapbook disk created by the Disk Manager will contain a Help page unless it is removed.

Help is really a special name for a page. Any page named Help will become the Help page.

If the **help** keys have been used to display the Help page:

- You can't go down to the command center.
- You can't flip the page.
- Pressing Escape returns you to the page you were on previously.

To edit the Help page, choose it from Contents or get it with the **getpage** command. It is then a regular page. Any changes you make to the page will be saved when you return to Contents or go to another page.

Startup

Startup is a special name which can be given to any page. When LogoWriter starts up, it automatically goes to the Startup page right away, bypassing Contents. The features of a Startup page are described in Appendix 2, *Startup*.

Changing Scrapbook Disks

At any time while using LogoWriter, you can change scrapbook disks. If you change disks while a page is on the screen, the page will be saved on the new disk. To change disks without transferring pages:

- Press **Escape** to go to Contents.
- Change disks.
- Press **Escape** again, and Contents will list the pages on the new disk.

Note: Go to the Shapes page, or type the command **getshapes**, soon after changing scrapbooks. Otherwise, the shapes that LogoWriter uses will be the shapes from the previous scrapbook.

Copying Pages

Important: In Apple computers, the following copying techniques only work when you are *not* using subdirectories for storing pages. If you're using subdirectories, copy files with the **copyfile** command. **Copyfile** can copy any kind of file, including the Shapes page.

Copying A Page Onto Another Disk

Follow these steps to copy a page onto another disk.

- With the source disk in the disk drive, go to the page you want to copy.
- Remove the source disk and insert the destination disk - the disk onto which you want the page copied.
- Press **Escape**. The page is automatically saved on the destination disk.

Copying A Shapes Page Onto Another Disk

Follow these steps to copy a set of shapes onto another disk.

Note: The shapes on the destination disk will be replaced.

- With the source disk in the disk drive, go to the Shapes page. Make sure that you see the shapes that you want to copy.
- Remove the source disk and insert the destination disk - the disk onto which you want the shapes copied.
- Press **Escape**. The shapes are automatically saved on the destination disk.

Copying Pages With Two Disk Drives

Type these instructions to copy a page from a disk in drive A to another disk in drive B.

Note: These instructions will not work on a COMMODORE 64 computer, since there is only one disk drive.

```
setdisk  "a        Set LogoWriter to drive A.
getpage  "----     Use the page name here.
setdisk  "b        Set LogoWriter to drive B.
newpage            By changing pages, you save
                   the page.
setdisk  "a        Reset LogoWriter to drive A.
```

The following procedures can be used to copy every page from a scrapbook in drive A to a scrapbook in drive B. They will not work if any of the pages contain a procedure named **startup**, since a **startup** procedure will take control and change the page before it is copied.

```
to  copypages
setdisk  "a
copyall  pagelist
end

to  copyall  :pages
if  empty?  :pages  [stop]
getpage  first  :pages
setdisk  "b
newpage
setdisk  "a
copyall  bf  :pages
end
```

II LogoWriter Keys

Special keys and key combinations provide an easy way of getting around LogoWriter; shortcuts are available for many graphic and word processing features. This is a complete listing of LogoWriter's keys and their functions. Most of these keys have corresponding commands.

On different computer keyboards, some of these functions are assigned to different keys. For a comparison of the different keyboards, see Section 3, *LogoWriter Versions*.

General Purpose Keys

Flip Keys

The **flip** keys flip the page from the front side to the flip side and back.

Up Keys

The **up** keys activate the cursor on the page for typing in text (on the front side) and for typing in procedure definitions (on the flip side).

Down Keys

The **down** keys activate the cursor in the command center for typing in LogoWriter instructions.

Escape

The **Escape** key is used to leave a page or a special mode. If a *special mode* has been entered (by pressing the **select, help**, or **turtle-move** keys, for example), pressing **Escape** will leave the mode, returning you to wherever you were before. If you are not in a mode, pressing **Escape** will leave the page, returning you to Contents.

Note: The COMMODORE 64 computer has no key labeled **Escape**. Instead, the F1 key functions as the **Escape** key.

Open a Line Keys
The **open a line** keys insert a blank line starting at the cursor.

Stop Keys
Pressing the **stop** keys stops any program or instruction which is running, returning you to the command center.

Help Keys

Help Keys
The **help** keys get the Help page.

Escape
Escape leaves the Help page and returns you to the page you were on previously.

Graphics Keys

Turtle-Move Keys
The **turtle-move** keys start the turtle-move mode. In the turtle-move mode, only one function is available: the turtle can be moved around the screen with the **arrow** keys.

Label Keys
The **label** keys start the label mode. In this mode, only the labeling function is available: any text that is typed becomes part of the graphic picture. The **arrow** keys are used to move the label cursor around the page.To delete a typing error in label mode, put the cursor over the mistake and retype the same letter.

Escape
Pressing Escape leaves any mode.

Word Processing Keys

Select Keys

Pressing the **select** keys starts the select mode. Moving the cursor while in select mode "selects" the text that the cursor crosses over. The selected text is highlighted either by a change in color or by changing to inverse video characters, depending on the version of LogoWriter you're using. The selected text can then be cut or copied using a command or key combination.

Cut Keys

The **cut** keys remove the selected text, storing it in a section of the computer's memory called the Clipboard.

Copy Keys

The **copy** keys store a copy of the selected text in a section of the computer's memory called the Clipboard.

Paste Keys

The **paste** keys insert the text that is stored in the Clipboard, into the page, at the cursor position.

Escape

Pressing Escape while in select mode turns off the mode.

Erase to End of Line Keys

The **erase to end of line** keys erase all the text from the cursor position to the end of the line. This text is not stored in the Clipboard, and *cannot* be recovered.

Next Screen Keys

The **next screen** keys display the next screenful of text on the page (if there is any).

Previous Screen Keys

The **previous screen** keys display the previous screenful of text on the page (if there is any).

Top of Page Keys

The **top of page** keys move the cursor to the very beginning of the text on the page.

Bottom of Page Keys

The **bottom of page** keys move the cursor to the very bottom of the text on the page.

Beginning of Line Keys

The **beginning of line** keys move the cursor to the beginning of the line that the cursor is on.

End of Line Keys

The **end of line** keys move the cursor to the end of the line that the cursor is on. The "end of the line" is where the next carriage return character is. If several lines were typed without pressing Return Enter the end of the line may be several screen lines down.

Shapes Page Keys

Most of the Shapes page key functions are similar to the functions they have on a regular page.

Escape

Pressing Escape leaves the Shapes page, returning you to where you were when the Shapes page was chosen.

Flip Keys

The **flip** keys flip from the front side, where all the shapes on the page are displayed, to the flip side, where each shape is displayed one at a time for editing.

Next Screen Keys

The **next screen** keys display the next shape for editing.

Previous Screen Keys

The **previous screen** keys display the previous shape for editing.

Arrow Keys

The **arrow** keys move the shape cursor around in the shape.

Space Bar

Pressing the **space bar** empties or fills a square within a shape.

Cut Keys

The **cut** keys clear a shape, storing the shape temporarily in the computer's memory.

Copy Keys

The **copy** keys store a copy of a shape temporarily in the computer's memory, without erasing it.

Paste Keys

The **paste** keys put the shape that has been placed in memory (by a **cut** or a **copy**) into the shape that is displayed.

Stop Keys

Pressing the **stop** keys cancels editing, and restores the shape that was there before editing began.

Contents Keys

Return / Enter

Pressing **Return** **Enter** chooses the page (or the subdirectory) that the cursor is on.

Escape

If the scrapbook contains only pages (and no subdirectories), pressing **Escape** redisplays the pages on the disk. If subdirectories are being used, pressing **Escape** backs up one level of subdirectory and displays the Contents page for the higher level directory.

Up and Down Arrow Keys

The **up** and **down arrow** keys move the cursor up and down through the page names that are on the scrapbook.

Top of Page Keys

The **top of page** keys move the cursor to the top of the Contents list (New Page).

Bottom of Page Keys

The **bottom of page** keys move the cursor to the last page name in the Contents list.

Previous Screen Keys

The **previous screen** keys display the previous screenful of page names (if there is one).

Next Screen Keys

The **next screen** keys display the next screenful of page names (if there is one).

Erase to End of Line Keys

Pressing the **erase to end of line** keys erases the page named on the line that the cursor is on.

Warning: Erasing a page is permanent. An erased page cannot be recovered.

Section 2: LogoWriter Primitives

This section consists of a description of each LogoWriter primitive.

Each primitive is described in the same way.

Here is an example of how a primitive is described:

---

① **forward** *number* ② ④ Command/1 input ⑤

③ **fd** *number*

⑥ Moves the turtle or turtles forward a distance of *number* steps.

Examples:

⑦
```
rg
fd  50
rt  90
repeat  50  [fd  2]
```

---

① Primitive name.

② Type of input.

③ Short form.

④ Type of primitive (command, reporter, or special word).

⑤ Number of inputs most commonly used.

⑥ Definition.

⑦ Examples of use.

Here is an example of how the top line of a primitive with more than one choice of inputs would look:

print word/list Ⓐ(**print** word/list1 word/list2...) **pr** word/list	Command/1 input If inside parentheses, one or many inputs.

Ⓐ Shows that any number of inputs can be used. If you use a different number of inputs from the number most commonly used, the primitive and its inputs must be enclosed in parentheses.

Input Words

On the following pages you will find an alphabetical listing of the types of inputs that LogoWriter primitives use. Beside each word is a description of what the actual input must be.

Type of Input	Real Input
address	A number representing a memory location within the computer: a decimal integer from 0 through the number of bytes of RAM memory the computer contains. In a 128K computer the highest number is 131072.
buttonnumber	An integer: 0, 1, 2, or 3.
byte	A unit of data used by the computer: a decimal integer from 0 through 255.
char	A letter of the alphabet, number, or punctuation mark.

degrees	Degrees of an angle: a number between -10000 and 10000.
frequency	A number from 37 to 9999.9999.
letter	A letter of the alphabet.
list	Information enclosed in brackets [ ].
listtorun	A list of procedures or primitives that LogoWriter can run.
name	A word naming a procedure or a variable.
namelist	A list of names.
number	A number between -10000 and 10000.
pagename	The name of a LogoWriter page.
pathname	The name representing the path from the main directory of a disk to a file on the disk, including the names of the subdirectories containing the file.
time	A number from 0 to 255: time specified in ticks, each tick equal to approximately one 20th of a second.

true/false	A primitive or procedure that reports the word **true** or the word **false**.
turtle	An integer from 0 to 3.
turtlelist	A list of any of the integers from 0 to 3.
word	A sequence of characters (not including spaces). A number is a type of word.
word/list	A word or a list.
[x y]	A list of two numbers giving the coordinates of the turtle.

Infix Math Reporters

number1 + *number2* (Addition) Reporter/2 inputs

Reports the sum of *number1* and *number2*.

Example:

```
show 5 + 3
8
```

number1 - *number2* (Subtraction) Reporter/2 inputs

Reports the difference between *number1* and *number2*.

Example:

```
show 8 - 3
5
```

number1 * *number2* (Multiplication) Reporter/2 inputs

Reports the product of *number1* multiplied by *number2*.

Example:

```
show 2 * 3
6
```

number1 / *number2* (Division) Reporter/2 inputs

Reports the quotient resulting when *number1* is divided by *number2*.

Example:

```
show 12 / 3
4
```

word/list1 = *word/list2* (Equal to) Reporter/2 inputs

Reports **true** if *word/list1* and *word/list2* are equal numbers *or* the same words or lists; otherwise, it reports **false**. Two words are considered equal even if one uses a capital letter and the other doesn't.
See **equal?** and **identical?**

Example:

```
show 8 = 7
false
show 4 * 4 = 16
true
show "hello = "HELLO
true
show "hello = [hello]
false
```

number1 > *number2* (greater than) Reporter/2 inputs

Reports **true** if *number1* is greater than *number2*; otherwise, it reports **false**.

Example:

```
show 8 > 3
true
```

number1 < *number2* (less than) Reporter/2 inputs

Reports **true** if *number1* is less than *number2*; otherwise, it reports **false**.

Example:

```
show 8 < 3
false
```

A

all Reporter/No inputs

Reports the list of turtles [0 1 2 3].
See **tell** and **ask**.

Example:

```
rg
tell  all
st
fd  50
show  who
[0  1  2  3]
```

and *true/false1 true/false2*
(**and** *true/false1 true/false2 true/false3...*)
Test Reporter/2 inputs
If inside parentheses, one or many inputs.

Reports **true** if *all* of its inputs report **true**; otherwise, it reports **false**.

Examples:

```
show  and  "true  "false
false
show  and  (5  =  5)    ("A  =  "A)
true
```

LogoWriter 2.0 only.

Reports the arc tangent (the inverse function to the tangent) of *number*. The number reported is a decimal number and is in degrees, not radians. **Arctan** always reports a number between -90 and 90.

Examples:

The following procedures use **arctan** and **sqrt** to create the functions **arcsin** and **arccos**.

```
to arcsin :x
if :x = 1 [op 90]
if :x = -1 [op -90]
output arctan :x / (sqrt 1 - :x * :x)
end

to arccos :x
if 0 = :x [op 90]
output arctan (sqrt 1 - :x * :x) / :x
end

show sin 30
0.5
show arcsin .5
30.0012
show arccos .7071
45.0028
```

ascii char — Reporter/1 input

Reports the ASCII number which represents the character.

Examples:

```
show ascii "a
97
show ascii "b
98
show ascii "A
65
```

ask *turtle listtorun* — Command or Reporter/2 inputs
ask *turtlelist listtorun*

Temporarily addresses the turtle or turtles named, giving them the instructions in the instruction list. This does not change the list of turtles which was set by the last **tell** command.

If *listtorun* is a reporter, **ask** reports whatever the reporter outputs.
See **tell**.

Examples:

```
rg
tell all st
rt 90
fd 50
ask 0 [fd 50]
rt 90
fd 50
setpos ask 0 [pos]
```

B

back *number* Command/1 input
bk *number*

Moves the turtle or turtles backwards a distance of *number* steps.

Examples:

```
rg
back  50
rt  90
back  50
```

bg Reporter/No inputs

Stands for **b**ack**g**round. Reports the number representing the background color.
See **setbg**.

Example:

```
rg
setbg  3
show  bg
3
rg
```

bottom Command/No inputs

Moves the cursor on the page to the bottom of the text.

Example:

```
ct
print [hello there]
hello there

insert [how are you?]
how are you?
top
hello there
bottom
how are you?
```

butfirst *word/list* Reporter/1 input
bf *word/list*

Reports all but the first element of the word or list.

Examples:

```
show bf "hello
ello
show bf [one two 3 4 five]
[two 3 4 five]
```

butlast *word/list* Reporter/1 input
bl *word/list*

Reports all but the last element of the word or list.

Examples:

```
show bl "goodbye
goodby
show bl [one two 3 4 five]
[one two 3 4]
```

button? *buttonnumber* Test Reporter/1 input

Reports **true** if the button *buttonnumber* on a game paddle (or joystick plugged into the game port) is pressed; otherwise, it reports **false**.

Examples:

```
show button? 1
false
```

Define this procedure on the flip side.

```
to check
if button? 1 [stop]
fd 1
check
end
```

In the command center, type:

```
check
```

Press the button on paddle 1 or button 1 on the joystick.

Note: On the COMMODORE 64 computer, the input to **button?** should be 2. You should only use a joystick in socket (port) 2.

C

cb — Command/No inputs

Stands for **c**ursor **b**ack. Moves the cursor on the page back one character. This is similar in function to the **left arrow** key.

Example:

```
ct
print  "hello
hello
□
cb  cb
hell◎
```

cc — Command/No inputs

Stands for **c**lear **c**ommands. Clears all text from the command center.

cd — Command/No inputs

Stands for **c**ursor **d**own. Moves the cursor on the page down one line. This is similar in function to the **down arrow** key.

Example:

```
ct
pr  "hello  pr  "there
hello
there
□
top
hello
there
cd
hello
there
```

cf — Command/No inputs

Stands for **c**ursor **f**orward. Moves the cursor on the page forward one character. This is similar in function to the **right arrow** key.

Example:

```
ct
pr  "hello
hello
□
top
hello
cf  cf
hello
```

cg Command/No inputs

Stands for **c**lear **g**raphics. Clears all graphics and sends the turtle or turtles to their home positions pointing upwards. Graphics are all lines and shapes drawn, stamped or filled, and all text printed with the **label** command or in label mode.
See **clean** and **rg**.

Example:

```
rg
fd 50
setsh 21
label "hello
pr "there
cg
```

char *number* Reporter/1 input

Reports the text character with the ASCII code of *number*.

Examples:

```
show char 65
A
show char 32
```

An empty space is printed.

```
show char 97
a
```

charunder — Reporter/No inputs

Reports the letter that is underneath the turtle. **Charunder** only finds characters that are typed in regular text, not characters typed with the **label** primitive or in the label mode. If there is no letter under the turtle, the space character is reported.

Example:

```
ct
print  "hello
hello
□
top
setpos  cursorpos
show  charunder
h
```

clean — Command/No inputs

Erases all the graphics from the page without affecting the turtles or the text on the page in any way. Graphics are all lines and shapes drawn, stamped or filled, and all text printed with the **label** command or in the label mode. See **cg** and **rg**.

Example:

```
rg
fd  50
clean
```

clearevents Command/No inputs

Clears all programmed control-key events from memory. Control-key events are programmed with the **when** command.
See **when**.

Example:

```
when "z [print [I'm sleeping]]
```

Press Control-Z

```
I'm sleeping
clearevents
```

Press Control-Z

Nothing happens.

clearname *word/list* Command/1 input

Clears the variable name or names and their values from memory. Variables are created with the **make** and **name** primitives.
See **make** and **name**.

Example:

```
name "Bob "friend
show :friend
Bob
clearname "friend
show :friend
friend has no value
```

clearnames — Command/No inputs

Clears all variable names and their values from memory. Variables are created with the **make** and **name** primitives. See **make** and **name**.

Example:

```
name "Bob "friend
make "class [Peter Anne Dennis Geni]
show :friend
Bob
clearnames
show :friend
friend has no value
show :class
class has no value
```

cleartools — Command/No inputs

Clears all "tool" procedures from memory. Tool procedures are LogoWriter procedures which have been loaded into memory with the **gettools** primitive. They are not shown on the flip side, and are not erased when a page is changed. **Cleartools** is the only way of clearing tools See **gettools** and Appendix 3, *Tools*.

clipboard — Reporter/No inputs

Reports as a word the contents of the text buffer called the "clipboard." If the clipboard is empty, **clipboard** reports the empty word.

Example:

```
pr [Today is Tuesday.]
Today is Tuesday.

top
select
eol
Today is Tuesday.
cut
show clipboard
Today is Tuesday.
```

color — Reporter/No inputs

Reports the number representing the color of the turtle (which is the color it draws in). If more than one turtle is active, **color** gives the color number for the first turtle in the list output by **who**.
See **setc**.

Example:

```
rg
setc 2
fd 50
show color
2
```

colorunder Reporter/No inputs

Reports the number representing the color under the center of the turtle, where its pen is located. If the pen is over a drawn line, the color that **colorunder** reports will be the color of the line rather than the color the turtle seems to be over.

Examples:

```
rg
setbg  0
show  colorunder
0
setc  3
fd  50
show  colorunder
3
pu
fd  5
show  colorunder
0
```

contents Command/No inputs

Saves the current page, then displays the special page Contents which lists all pages on the scrapbook disk. Contents acts as a menu from which any page can be accessed or erased.

copy — Command/No inputs

Puts a copy of the selected text into a text buffer called the "clipboard." Text must first have been selected with the **select** command or the **select** keys for it to be copied. See **select**, **cut**, **clipboard**, and **paste**.

Example:

```
ct
pr [Bob is great]
Bob is great
□
top
Bob is great
select
cf cf cf
Bob is great
copy
Bob□is great
cd
paste
Bob is great
Bob
□
```

cos *degrees* — Reporter/1 input

Stands for **cos**ine. Reports the cosine of *degrees*.

Examples:

```
show cos 45
0.7071
seth 90
show cos heading
0
```

count *word/list* Reporter/1 input

Reports the number of elements in the word or list. If the input is a word, **count** reports the number of letters in the word. If the input is a list, **count** reports the number of items (words or lists) in the list.

Examples:

```
show count "hello
5
show count [one one one]
3
show count []
0
```

cp Command/No inputs

Stands for **c**lear **p**age. Clears all text and graphics from *both sides* of the page.

Note: This erases all procedures as well as clearing the front of the page. If a page is accidentally cleared, the **restore** command will bring back the version of the page which is on the disk. The **undo** command brings back the text that was cleared if it is used immediately after **cp**. See **restore** and **undo.**

In LogoWriter 2.0, **cp** goes to a new clean page. This is an extra protection against accidentally erasing a page's contents.

Example:

```
newpage
```

Define this procedure on the flip side.

```
to square
repeat 4 [fd 50 rt 90]
end
```

In the command center, type:

```
square
cp
square
I don't know how to square
```

---

ct Command/No inputs

Stands for **c**lear **t**ext. Clears all the text from the side of the page that is displayed. Does not affect any text that has been typed in the label mode or with the **label** command.

If text is accidentally cleared, the **undo** command brings back this text if it is used immediately after **ct**.

Note: Clearing the text on the flip side of the page is one way to erase all of the procedures on the page.

Example:

```
print "hello
hello
□
ct
```

cu — Command/No inputs

Stands for **c**ursor **u**p. Moves the cursor on the page up one line. This is similar in function to the **up arrow** key.

Example:

```
ct
print "hello pr "there
hello
there
cu
hello
there
```

cursorchar — Reporter/No inputs

LogoWriter 2.0 for IBM computers only.

Reports the character which is directly under the cursor on the page. **Cursorchar** can only read text characters, not characters printed with **label**.

Example:

```
ct
print "hello
top
repeat 5 [show cursorchar cf]
h
e
l
l
o
```

cursorpos — Reporter/No inputs

Reports the position of the cursor on the page, in turtle coordinates.

Examples:

```
ct
repeat 2 [pr [Where am I?]]
Where am I?
Where am I?

top
Where am I?
Where am I?
setpos cursorpos
bottom
setpos cursorpos
```

cut — Command/No inputs

Cuts the selected text from the page and puts it in a text buffer called the "clipboard." Text must first have been selected with the **select** command or the **select** keys. See **select**, **copy**, **clipboard**, and **paste**.

Example:

```
ct
print [think quick]
think quick

top
think quick
select
repeat 5 [cf]
think quick
cut
quick
```

D

delete — Command/No inputs

Deletes the character under the cursor on the page.

Example:

```
ct
print  "hello
hello
□
top
delete
ello
```

disk — Reporter/No inputs

On a computer with more than one disk drive, reports the letter representing the current disk drive. LogoWriter 2.0 for Apple computers can also set LogoWriter to the slot which is connected to the disk drives.
See **slot** and **setslot** in the chapter *ProDOS Primitives*, and **setdisk**.

Example:

```
show  disk
A
```

distance *[x y]* Reporter/1 input

LogoWriter 2.0 for IBM computers only.

Reports the distance in turtle steps from the turtle to the screen position *[x y]*.

Examples:

```
rg
seth towards [70 70]
fd distance [70 70]
seth towards [-15 90]
fd distance [-15 90]
```

dspace Command/No inputs

Stands for **d**ouble **space**. Sets the printer to print double spaced text when **printtext** or **printtext80** is used. The setting will remain until the **sspace** command sets it to print single spaced text.
See **sspace**, **printtext**, and **printtext80**.

Example:

```
repeat 10 [print [one two three]]
sspace
printtext
dspace
printtext
```

E

each *listtorun* Command/1 input

Makes each turtle that is following commands run the instructions in *listtorun*. If there is more than one active turtle, the first turtle runs all the instructions in *listtorun* before the second turtle does anything, and so on. See **tell** and **ask**.

Examples:

```
rg
tell [0 1 2 3]
st
each [repeat 4 [fd 20 rt 90]]
tell [1 3]
each [seth (90 * who) fd 80]
```

empty? *word/list* Test Reporter/1 input

Reports **true** if *word/list* is an empty word or an empty list; otherwise, it reports **false**.

Examples:

```
show empty? [ ]
true
show empty? "
true
show empty? [0]
false
```

end — Special Word

Indicates the end of a procedure definition and separates procedures from one another on the flip side of the page. **End** must be alone on its own line at the end of a procedure.

Examples:

Define these procedures on the flip side.

```
to square
repeat 4 [fd 50 rt 90]
end
```

```
to triangle
repeat 3 [fd 30 rt 120]
end
```

eol — Command/No inputs

Stands for **e**nd **o**f **l**ine. Moves the cursor on the page to the end of the line (the next carriage return). This is similar in function to the **end of line** keys.

Example:

```
ct
print [hello there]
hello there
□
top
hello there
eol
hello there□
```

equal? *word/list1 word/list2* Test Reporter/2 inputs

Reports **true** if *word/list1* and *word/list2* are the same, even if one uses a capital letter and the other one doesn't; otherwise, it reports **false**. A word and a list containing that word are not equal.
See **identical?** and =.

Examples:

```
show equal? "a "A
true
show equal? "one 1
false
show equal? [hello] [hello]
true
show equal? "hello [hello]
false
```

erpage *pagename* Command/1 input

Stands for **er**ase **page**. Erases the page from the scrapbook on the disk. Use with caution - once a page is erased you cannot get it back.

Example:

```
erpage "tulips
```

Use the name of a page you want to erase.

F

false — Special Word

Used as input to **if**, **ifelse**, **and**, **or**, and **not**.

Examples:

```
show and "true "false
false
ifelse "false [show "yes] [show "no]
no
```

fastturtle — Command/No inputs

LogoWriter 2.0 for IBM computers only.

Sets the turtle's drawing speed. To slow the turtle's drawing speed, use **slowturtle**. When LogoWriter starts up, the turtle speed is set to **fastturtle**.

fill — Command/No inputs

Fills a shape or the entire screen with the turtle's color. **Fill** starts drawing at the position of the pen; if the pen is over a drawn line, only the line is filled. In order to fill a shape with a color, the shape must be closed: pick the pen up, put the turtle inside the area to be filled, put the pen down, and fill.

Example:

```
rg
repeat 4 [fd 50 rt 90]
pu
rt 45 fd 10
pd
fill
```

first *word/list* Reporter/1 input

Reports the first element of *word/list*. **First** of a word is a single character; **first** of a list can be a word or a list.

Examples:

```
show first "hello
h
show first [three two one go]
three
```

flip Command/No inputs

Causes the page to flip from the front side to the flip side or from the flip side to the front side. Similar in function to the **flip** keys.

Examples:

```
flip
if not front? [flip]
```

forward *number* Command/1 input
fd *number*

Moves the turtle or turtles forward a distance of *number* steps.

Examples:

```
rg
fd 50
rt 90
repeat 50 [fd 2]
```

found? Test Reporter/No inputs

Used after **search** or **replace**. Reports **true** if the desired word was found by **search** or **replace**; otherwise, it reports **false**.

Example:

```
ct
pr "hello
hello
□
top
search "o
if found? [insert "ooo]
hellooooo□
```

fput *word/list list* Reporter/2 inputs

Stands for **f**irst **put**. Reports a new list which is formed by putting *word/list* at the beginning of *list*.

Examples:

```
show fput "i [do it]
[i do it]
show fput [i] [do it]
[[i] do it]
```

front? Test Reporter/No inputs

Reports **true** if the front side of the page is displayed; otherwise, it reports **false**.

Examples:

```
show front?
true
if front? [flip]
```

G

getpage *pagename* Command/1 input
gp *pagename*

Saves the page that is showing, and displays *pagename*. Any procedures on *pagename*'s flip side can be used. If *pagename* does not exist, a message is displayed.

Example:

```
getpage  "hello
```

getshapes Command/No inputs

LogoWriter 2.0 only.

Replaces the shapes in memory with the shapes on the current Shapes page. **Getshapes** is a useful command to include in programs that change directories or change disks. You can use it to set LogoWriter's shapes to the shapes in the new directory or on the new disk.

If there is no Shapes page in the directory, the **getshapes** command is ignored.

Example:

The following example switches disk drives, then sets the shapes to the shapes on the disk in the second drive.

```
setdisk  "b
getshapes
```

Loads the procedures on the flip side of *pagename* into memory as tool procedures. Tool procedures do not appear on the flip side of the page that is showing. Tool procedures function like primitives: they can be used, but they are not visible on the flip side, and they are not erased when the page is changed.

If a procedure that is being loaded as a tool has the same name as one on the flip side of the page that is showing, it will not be loaded.
See **cleartools**, **toollist**, and Appendix 3, *Tools*.

Example:

```
newpage
gettools  "polygons
```

heading — Reporter/No inputs

Reports the turtle's heading in degrees, from 0 to 359.9999. If more than one turtle is active, **heading** reports the heading of the first turtle in the list reported by **who**.

Examples:

```
rg
show  heading
0
rt  45
show  heading
45
if  heading  =  45  [show  "northeast]
northeast
```

home — Command/No inputs

Moves each of the turtles to its own home position and sets its heading to 0.

Example:

```
rg
tell  all
st  pu
rt  45
fd  50
home
```

Stands for **h**ide **t**urtle. Makes the active turtle or turtles invisible. The turtle usually draws more quickly when it is hidden.
See **st**.

Example:

```
repeat 4 [fd 50 rt 90]
ht
```

identical? *word/list1 word/list2* Test Reporter/2 inputs

Reports **true** if the two words or lists are equal numbers, identical words, or identical lists; otherwise, it reports **false**. A word and a list containing the same word are not identical. Corresponding characters of each input must match in terms of uppercase and lowercase characters. See **equal?** and =.

Examples:

```
show identical? "a "A
false
show identical? [one] [one]
true
```

if *true/false listtorun* Command/2 inputs

The first input, *true/false*, must be a condition that **if** tests to be **true** or **false**. If *true/false* is **true**, *listtorun* is run; if *true/false* is **false**, *listtorun* is ignored.

Example:

```
if 25 + 25 = 50 [show [that's right]]
[that's right]
```

ifelse *true/false listtorun1 listtorun2*
Command or Reporter/3 inputs

The first input, *true/false*, must be a condition that **ifelse** tests to be **true** or **false**. If *true/false* is **true,** *listtorun1* is run; if *true/false* is **false**, *listtorun2* is run. If *listtorun1* or *listtorun2* reports (outputs), **ifelse** reports the same thing.

Examples:

ifelse as a command:

```
ifelse 100 < 50 [show "less][show
"more]
more
```

ifelse as a reporter:

```
show ifelse 100 < 50 ["less] ["more]
more
```

insert *word/list*
(**insert** *word/list1 word/list2...*) Command/1 input
If inside parentheses, one or many inputs.

Inserts a word or a list in the page at the cursor position without a carriage return (the cursor does not move to the next line).
See **print**, **show**, **type**, and **label**.

Examples:

```
ct
insert "hello insert [hello there]
hellohello there□
ct
print "hello insert [hello there]
hello
hello there□
```

int *number* Reporter/1 input

LogoWriter 2.0 only.

Reports the integer portion of the number given as input. The decimal part of the number is removed.
See also **round**.

Examples:

```
show int 5.1
5
show int 5.8
5
show int -5.7
-5
```

item *number word/list* Reporter/2 inputs

Reports the *number*th element of the word or list.

Examples:

```
show item 4 [which word is the fourth]
the
show item 2 "hello
e
```

key? — Test Reporter/No inputs

Reports **true** if a key has been pressed on the keyboard; otherwise, it reports **false**. The key must not have been read (by **readchar**, **readlist,** or **readlistcc**) in order for **key?** to report **true**. **Key?** does not remove the character from the key buffer; the character will remain unless it is read with **readchar**, **readlist**, or **readlistcc**.
See **readchar**, **readlist**, and **readlistcc**.

Examples:

```
show  key?
false
wait  100  show  key?
```

Immediately after pressing Return Enter, press a key.

```
true
a
```

...or whatever key you pressed.

The following procedure clears the key buffer. It is useful in programs where the key buffer may contain previously typed characters.

```
to  clearbuff
if  not  key?  [stop]
make  "x  readchar
clearbuff
end
```

label *word/list* Command/1 input

Prints the specified word or list as graphics, starting from each turtle's position. Text printed in this way is not part of the regular text: it doesn't scroll and cannot be edited with the text editing keys or primitives. Text that is printed with **label** can be erased by either labeling the exact same letters again, or by clearing the graphics with **cg**, **clean**, or **rg**. See **print** and **insert**.

Examples:

```
rg
pu
label  "hello
bk  10
label  "there
cg
tell  [0  1]
label  "hello
```

last *word/list* Reporter/1 input

Reports the last element of *word/list*.

Examples:

```
show  last  "hello
o
show  last  [three  two  one  go]
go
```

leavepage — Command/No inputs

LogoWriter 2.0 only.

Leaves the page and goes to Contents *without saving the page*. Use this command to leave a page if you don't want the original version of the page to be changed.

Examples:

```
newpage
fd 50
```

Press the Escape key.

```
Please name this page
leavepage
```

In the example below, a **choose** procedure asks whether a user wants to leave the page or not.

```
to choose
print [Press Y to go back to Contents.]
if readchar = "y
     [leavepage]
end
```

left *degrees* — Command/1 input
lt *degrees*

Turns the turtle or turtles *degrees* to the left.

Example:

```
lt 90
```

list *word/list1 word/list2*
(**list** *word/list1 word/list2 word/list3...*)

Reporter/2 inputs
If inside parentheses, one or many inputs.

Reports one list made up of *word/list1* and *word/list2*.

Examples:

```
show list "hello "there
[hello there]
show list [hello] [there]
[[hello] [there]]
```

list? *word/list* — Test Reporter/1 input

Reports **true** if *word/list* is a list; otherwise, it reports **false**.

Examples:

```
show list? [hello there]
true
show list? "hello
false
show list? [ ]
true
```

load *pagename* — Command/1 input

Loads the flip side of *pagename* onto the flip side of the page that is showing. The procedures from *pagename* are added at the end of any procedures that are already there.

Example:

```
load "tulips
```

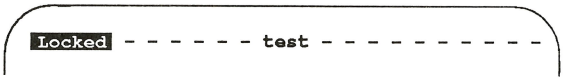
LogoWriter 2.0 only.

Lock protects a page from being erased or changed. You can look at a locked page, but can't save any changes. It is a good idea to lock a page containing a finished project. When you type **lock**, the page is saved as it is, and the word "LOCKED" appears beside the page name at the top of the screen.

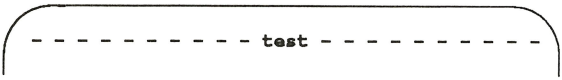
Pressing the Escape key on a locked page will NOT save any changes you have made. To save any changes, change the page name with **namepage** and then press the Escape key, or unlock the page with **unlock**.
See **unlock**.

Example:

```
namepage  "test
lock
```




```
unlock
```



lput *word/list list* Reporter/2 inputs

Stands for **l**ast **put**. Reports a new list which is formed by putting *word/list* at the end of *list*.

Examples:

```
show lput "it [I do]
[I do it]
show lput [it] [I do]
[I do [it]]
```

make *name word/list* Command/2 inputs

Creates a variable, *name*, and gives it the value *word/list*. In general, variables created with **make** are "global" and remain in the computer's memory even when the page is changed. The only exception is when a **make** instruction inside a procedure uses the same name as a variable on the title line of the procedure. In this case, the variable's value is "local," in memory only while the procedure is running.

Variables are not saved on a page unless the **make** instructions are put inside procedure definitions. Once the variable is created, the value can be accessed by :*name* or **thing** "*name*. The : (colon) means "the value of." See **name**, **clearname**, and **clearnames**.

Example:

```
make "class [Peter Anne Dennis Geni]
show :class
[Peter Anne Dennis Geni]
```

To save variables, put them in procedures on the flip side.

```
to savevariables
make "numbers [0 1 2 3 4 5 ]
make "message [hello there]
end
```

member? *word/list1 word/list2* Test Reporter/2 inputs

Reports **true** if *word/list1* is an element of *word/list2*; otherwise, it reports **false**.

Examples:

```
show member? "o "hello
true
show member? "there [hello there]
true
show member? "o [hello there]
false
```

minus *number* Reporter/1 input

LogoWriter 2.0 only.

Reports the additive inverse of (i.e. reverses the sign of) *number*.

Examples:

```
show minus 5
-5
show minus minus 5
5
seth -90
seth minus heading
show heading
90
```

name *word/list name* Command/2 inputs

Creates a variable, *name*, and gives it the value *word/list*. In general, variables created with **name** are "global," and remain in the computer's memory even when the page is changed. The only exception is when a **name** instruction inside a procedure uses the same name as a variable on the title line of the procedure. In this case, the variable's value is "local," in memory only while the procedure is running.

Name's function is identical to **make**, but the inputs are in the reverse order. Variables are not saved on a page unless the **name** instructions are put inside procedure definitions.
See **make**, **clearname**, and **clearnames**.

Example:

```
name  0  "black
setbg  2
setbg  :black
show  bg
0
```

name? *word* Test Reporter/1 input

Reports **true** if *word* is the name of a variable (if *word* has a value); otherwise, it reports **false**.
See **name** and **make**.

Examples:

```
make  "myname  "Sam
name  "Cathy  "friend
show  name?  "myname
true
show  name?  "friend
true
```

namepage *pagename* Command/1 input
np *pagename*

Names the page that is showing.

Example:

```
newpage
namepage  "tulips
```

newpage Command/No inputs

Saves the page that is showing, and displays a new page. The new page has no name (??? appears in the title position). The new page must be given a name with **namepage** before it can be saved.

Example:

```
newpage
namepage  "anyname
```

nextscreen Command/No inputs

Moves the cursor on the page to the next screenful of text. Similar in function to the **next screen** keys.

not *true/false* Test Reporter/1 input

Reports the logical inverse of its input; if *true/false* is **true**, **not** reports **false**. If *true/false* is **false**, **not** reports **true**. The input to **not** must report either **true** or **false**.

Examples:

```
show not "true
false
show 1 = 2
false
show not 1 = 2
true
```

number? *word/list* Test Reporter/1 input

Reports **true** if its input is a number; otherwise, it reports **false**. A number is a kind of word.

Examples:

```
show number? 12
true
show number? [22]
false
show number? 22 / 3
true
```

---

or *true/false1 true/false2*
(**or** *true/false1 true/false2 true/false3...*)

Test Reporter/2 inputs
If inside parentheses, one or many inputs.

Reports **true** if *any* of its inputs report **true**; otherwise, it reports **false**. **Or**'s inputs must report **true** or **false**.

Examples:

```
show or "true "false
true
show or (5 = 3) (2 = 2)
true
```

---

output *word/list* Command/1 input
op *word/list*

Stops the running procedure and reports *word/list*. **Output** can only be used within a procedure. It is used to create procedures which are reporters.

Example:

Define this procedure on the flip side.

```
to squared :number
output :number * :number
end
```

In the command center, type:

```
show squared 10
100
```

paddle *number* Reporter/1 input

Reports a number which indicates the position of the dial on game paddle *number* or the position of stick *number* on a joystick.

Paddle reports different numbers for different brands and types of computers. For more information on paddle and joystick use, see Section 3, *LogoWriter Versions*.

Example:

```
show paddle 0
44
```

The number will vary, depending on the paddle position.

pagelist Reporter/No inputs

Reports a list of the names of all the LogoWriter pages on the disk. In LogoWriter 2.0, a disk can also contain files that are not LogoWriter pages. To list these "hidden" files, use **filelist**.
See **filelist** in the chapter *ProDOS Primitives* (for Apple *II* computers) and the chapter *IBM DOS Primitives* (for IBM computers).

Example:

```
show pagelist
[tulips ladder pun garden]
```

Converts a string of characters (such as is reported by **selected** or **clipboard**) into a list and then reports this list. The list can then be used by LogoWriter's word and list processing primitives.

Examples:

```
ct
insert [This is an example of how parse
works.]
top select bottom
show parse selected
[This is an example of how parse
works.]
```

Parse also separates the list into words, following LogoWriter's internal rules for characters that "delimit," or act as separators.

```
ct
```

Press the **up** keys, and type:

```
3+3
```

Press the **down** keys.

```
top select bottom
show parse selected
[3 + 3]
```

There are spaces before and after the plus sign, because the plus sign is a delimiter.

paste — Command/No inputs

Inserts the text that is in the "clipboard" text buffer into the page at the cursor position.

Example:

```
ct
print [hello there]
hello there
□
top
select
repeat 5 [cf]
cut
□there
bottom
paste
 there
hello□
```

pd — Command/No inputs

Stands for **p**en **d**own. Puts the turtle's or turtles' pens down. When the turtle moves, it draws lines. Every turtle starts with its pen down.

Example:

```
rg
repeat 10 [pu fd 10 pd fd 10]
```

pe — Command/No inputs

Stands for **p**en **e**rase. Puts the turtle's or turtles' erasers down. When the turtle moves, it erases any lines it crosses.

Example:

```
rg
fd 50
pe
bk 50
```

pos — Reporter/No inputs

Stands for **pos**ition. Reports the coordinates of the turtle's position as a list [x y]. If there are several active turtles, **pos** will report the position of the first turtle in the list reported by **who.**

When you start LogoWriter, turtle 0 is at position [0 0]. See **setpos.**

Examples:

```
rg
show pos
[0 0]
tell 1
st
show pos
[0 -40]
```

prescreen — Command/No inputs

Stands for **pre**vious **screen**. Moves the cursor on the page to the previous screenful of text. Similar in function to the **previous screen** keys.

print *word/list*
(**print** *word/list1 word/list2...*) Command/1 input
pr *word/list* If inside parentheses, one or many inputs.

Prints the word or list on the page, followed by a carriage return. Outermost brackets of lists are not printed.
See **insert**, **show**, **type**, and **label**.

Examples:

```
ct
pr  "hello
hello

pr  [hello  there]
hello  there

pr  22  /  7
3.1429

(pr  [hello  there]  "Sam)
hello  there  Sam

```

printnames Command/No inputs

Prints all variables in memory on the page. Variables are created with **make** and **name**.
See **shownames**, **make**, and **name.**

Example:

```
make  "age  8
name  "Cathy  "friend
printnames
:friend  is  "Cathy
:age  is  8

```

printscreen Command/No inputs

Prints on a printer all the text and graphics that are *visible* on the page. Text in the command center and the title line of the page are not printed.

Note: If your printer does not respond properly, check the printer setting on the LogoWriter disk. If the printer specified is not compatible with the type you are using, change the printer setting on the disk.
See **printtext** and **printtext80**.

Example:

```
rg ct
repeat 4 [fd 50 rt 90]
pr "hello
label "there
printscreen
```

printtext Command/No inputs

Prints on a printer all the text on the current side of the page. This includes text which has scrolled above or below the screen. Text printed while in the label mode or with the **label** command is part of the graphics and is not printed. To print single spaced text, use the **sspace** command. To print double spaced printing, use **dspace**.

Note: If your printer does not respond properly, check the printer setting on the LogoWriter disk. If the printer specified is not compatible with the type you are using, change the printer setting on the disk.
See **printtext80**, **printscreen**, **sspace**, and **dspace**.

Example:

```
repeat 15 [pr "hello]
printtext
```

printtext80 — Command/No inputs

Prints on a printer, in 80-column mode, all the text on the side of the page that is showing. This includes text which has scrolled above or below the screen. Text printed while in the label mode or with the **label** command is part of the graphics and is not printed. To print single spaced text, use the **sspace** command. To print double spaced printing, use **dspace**.

Note: The appearance of the text when printed in 80 columns will be different from its appearance on the screen when it is in 40 columns. Be sure that carriage returns are used only at the end of a paragraph and that every paragraph ends with a carriage return.

If your printer does not respond properly, check the printer setting on the LogoWriter disk. If the printer specified is not compatible with the type you are using, change the printer setting on the disk.
See **printtext**, **printscreen**, **sspace**, and **dspace**.

Example:

```
ct
repeat 40 [insert "razamataz]
repeat 4 [print "hello]
printtext80
```

pu — Command/No inputs

Stands for **p**en **u**p. Puts the turtle's or turtles' pens up. When the turtle moves, it does not draw or erase.

Example:

```
rg
repeat 10 [pu fd 10 pd fd 10]
```

Puts the turtle's or turtles' reversing pens down. When the turtle moves, it draws lines where there aren't lines and erases where there are lines.

Example:

```
rg
px
repeat 10 [fd 50 bk 50 rt 9]
repeat 8 [fd 50 rt 90]
```

random *number* Reporter/1 input

Reports a random non-negative integer less than *number*.

Examples:

```
show random 10
4                          The result can be any
                           number smaller than 10.
fd random 100
```

readchar Reporter/No inputs

Stands for **read char**acter. Reads a character from the keyboard and then reports the character. If no character is waiting to be read, **readchar** waits until the user types something. The character is not printed on the screen or shown in the command center.
See **readlist**, **readlistcc**, and **key?**

Examples:

```
if readchar = "h [show "hello]     Type an h.
hello
show ascii readchar                Type an a.
97
```

Reads a line that is typed into the page, and reports it as a list. **Readlist** waits for the Return Enter key to be pressed before reading the line.

Note: Only characters typed in *after* the instruction using **readlist** are read. The **arrow** keys cannot be used to move the cursor onto text that was previously printed. See **readlistcc** and **readchar**.

Examples:

```
cc ct
show readlist
```
The cursor on the page is blinking. Type:

```
hello there
[hello there]
```
This appears in the command center.

```
pr "
insert [Type your name:]
make "myname readlist
Type your name:
Type your name:Terry Brown
```
Type your name.

Press Return Enter.

```
show :myname
[Terry Brown]
```
This appears in the command center.

Stands for **read list** from the **c**ommand **c**enter. Reads a line that is typed into the command center, and reports it as a list. **Readlistcc** waits for the the Return Enter key to be pressed before reading the line.

Note: Only characters typed in *after* the instruction using **readlistcc** are read. The **arrow** keys cannot be used to move the cursor onto text that was previously printed. See **readlist** and **readchar**.

Examples:

```
cc
show  readlistcc
hello  there
[hello  there]
```

This appears in the command center.

Define this procedure on the flip side.

```
to  question
type  [Type  your  name:]
make  "myname  readlistcc
end
```

In the command center, type:

```
question
type  your  name:Linda

show  :myname
[Linda]
```

Type your name and press Return Enter.

recycle Command/No inputs

Performs a "garbage collection" in the computer's memory, freeing up unused memory space. Adding a **recycle** command to large, complex programs can help them run more quickly, and can help prevent LogoWriter's automatic garbage collection from slowing down a program at the wrong time. Always use a **recycle** command before using **space** to find out the true percentage of available workspace.
See **space**.

Example:

```
recycle
show  space
```

remainder *number1 number2* Reporter/2 inputs

Reports the remainder, if any, after *number1* is divided by *number2*. If there is no remainder, **remainder** reports 0.

Examples:

```
show  remainder  12  5
2
show  remainder  10  5
0
```

repeat *number listtorun* Command/2 inputs

Runs the list of instructions *listtorun* the specified *number* of times.

Examples:

```
rg
repeat 4 [fd 50 rt 90]
repeat 5 [show [hello there]]
[hello there]
[hello there]
[hello there]
[hello there]
[hello there]
```

replace *word word/list* Command/2 inputs

Searches for *word* in the text on the page from the cursor position to the bottom of the text, and replaces the first occurrence of *word* with *word/list*. If the word isn't found, the cursor will not move.
See **search** and **found?**

Examples:

```
ct
pr [to be or not to be]
to be or not to be
□
top
replace "be "walk
to walk□or not to be
replace "be "walk
to walk or not to walk□
```

restore — Command/No inputs

Replaces the page that is showing with the version of that page that is on the disk. Any changes or editing done to either side of the page since it was first displayed will be lost. If the page is new (there is no version of it on the disk), there is no change.

Example:

```
getpage  "tulips
```

Use the name of one of your pages.

```
rg  fd  50
restore
```

The page comes back as it was.

rg — Command/No inputs

Stands for **r**eset **g**raphics. Resets the graphic system to the original state: all drawings are erased, the background is black (blue on a COMMODORE 64), the turtle is white and set to the turtle shape, the turtle's pen is down, and only turtle 0 is showing and following commands. The other three turtles are hidden and inactive.
See **cg** and **clean**.

Example:

```
fd  50
setsh  12
setc  2
tell  [0  1  2  3]
st
label  "there
rg
```

right *degrees* Command/1 input
rt *degrees*

Turns the turtle or turtles *degrees* to the right.

Example:

```
rg
rt  90
```

round *number* Reporter/1 input

LogoWriter 2.0 only.

Reports the number given as input, rounded to the nearest integer. See also **int**.

Examples:

```
show  round  5.1
5
show  round  5.8
6
show  round  -5.7
-6
```

run *listtorun* Command or Reporter/1 input

Runs *listtorun*. If *listtorun* reports, **run** reports the same thing.

Examples:

```
run  [fd  50]
show  run  [2  +  4]
6
make  "problem   [2  +  3  +  5  +  6  +  1]
show  run  :problem
17
```

savepage Command/No inputs

LogoWriter 2.0 only.

Savepage saves the current page without returning to Contents.

search *word* Command/1 input

Looks for the first occurrence of *word* in the text on the page starting from the cursor position and going to the bottom of the text. If *word* is found, whether as a whole word or as part of another word, it is selected. Use **unselect** to undo the highlighting effect of **search**. See **replace**, **found?**, and **unselect**.

Example:

```
ct
pr "hello
hello
□
top
hello
search "o
hello
if found? [insert "ooo]
hellooooo□
```

Starts the select mode. If the instructions that follow the **select** command move the cursor (for example, **cf**, **cb**, **cu**, **cd**, **sol**, **eol**), a block of text will be selected. The selected text extends from the starting cursor position to the cursor position created by the instructions. If **select** is followed by any text editing instructions that change the text (for example, **search**, **replace**, **print**) or by **unselect**, the select mode is turned off.
See **unselect**.

Example:

```
ct
pr  [hello  there]
hello  there
□
select
cb  cb  cb
hello  there
copy
bottom  paste
hello  there
re
□
```

selected Reporter/No inputs

Reports as a word a copy of the characters in the block of selected text on the page. The text may have been selected by key-operations (the **select** keys and **arrow** keys) or by instructions (**select** or **search**). If no block is selected, the empty word is reported. Use **parse** to convert the block of characters into a list.
See **select**, **search**, and **parse**.

Example:

```
ct
pr [hello there]
hello there

select
cb cb cb
hello there
show selected
re
```

sentence *word/list1 word/list2*
(**sentence** *word/list1 word/list2 word/list3...*)
se *word/list1 word/list2* Reporter/2 inputs
If inside parentheses, one or many inputs.

Reports a list which is made up of its inputs. If the inputs are lists, **sentence** combines them to make a single list.
See **list.**

Examples:

```
show se "hello "there
[hello there]
show se [hello] [there]
[hello there]
show se "hello [there]
[hello there]
```

setbg *number* Command/1 input

Stands for **set b**ack**g**round. Sets the background color of the screen to the color represented by *number*.
See **bg**.

Example:

```
rg
setbg  2
show  bg
2
```

setc *number* Command/1 input

Stands for **set c**olor. Sets the color of each turtle and its pen to the color represented by *number*.
See **color.**

Examples:

```
rg
setc  1  fd  40
setc  2  fd  40
show  color
2
```

setdisk *letter* Command/1 input

On a computer system with more than one disk drive, sets the drive for saving and loading pages. The first drive is drive A, the second is drive B or C, depending on the computer system.
See **disk**, and the chapters *IBM DOS Primitives*, *ProDOS Primitives* and the section *LogoWriter Versions*.

Examples:

```
setdisk  "b
contents
```

To copy a page from drive A to drive B:

```
setdisk  "a
gp  "tulips
```

Use one of your page names.

```
setdisk  "b
```

Press **Escape**.

seth *degrees* Command/1 input

Stands for **set h**eading. Sets the heading of the turtle or turtles to *degrees*.
See **heading**.

Examples:

```
rg
show  heading
0
seth  45
fd  50
seth  180
fd  50
```

setpos *[x y]* Command/1 input

Stands for **set pos**ition. Moves the turtle or turtles to the given screen position.
See **pos**.

Examples:

```
rg
setpos  [0  50]
setpos  [50  50]
setpos  [50  0]
home
```

setsh *number* Command/1 input

Stands for **set sh**ape. Sets the shape of the turtle or turtles to the shape specified by *number*.
See **shape**.

Examples:

```
rg
setsh  11
show  shape
11
setsh  shape  +  1
```

settc *number* Command/1 input

LogoWriter 2.0 for IBM computers only.

Stands for **set t**ext **c**olor. Changes the color of the selected text, or the next text to print, to the color *number*. If the cursor is moved it will take on the color of the text underneath.
See **tc**.

Examples:

```
ct  settc  1
print  "hello          Hello is printed in white.
settc 2  print "there  There is printed in pink.
top                    The cursor changes as it
                       passes over the white text.
```

settextpos *number* Command/1 input

Stands for **set text pos**ition. Sets the cursor position in the page to *number* characters from the top of the text.
See **textpos**.

Examples:

```
ct
pr [think quick]
think quick
□
settextpos 2
think quick
select
settextpos textpos + 6
think quick
cut
thick
```

setx *number* Command/1 input

LogoWriter 2.0 only.

Sets the x-coordinate (the horizontal placement) of the turtle to the position given by *number*. The y-coordinate (the vertical position) is not changed. The turtle's heading is not changed.
See **sety**, **xcor**, and **setpos**.

Examples:

```
rg
setx 50
fd 35
rt 30
setx -60
```

sety *number* Command/1input

LogoWriter 2.0 only.

Sets the y-coordinate (the vertical placement) of the turtle to the position given by *number*. The x-coordinate (the horizontal position) is not changed. The turtle's heading is not changed.
See **setx**, **ycor**, and **setpos**.

Examples:

```
rg
sety  -45
lt  90
fd  60
sety  100
```

shade Command/No inputs

Shades the page, or an enclosed area of the page, with stamped copies of the turtle's shape. The area that is shaded is the area in which the center of the turtle is located. **Shade** can't shade an area smaller than the turtle itself or an area with a large number of sides. To shade an area, first pick up the pen, then move the turtle into the area, put the pen down, and shade.

Example:

```
rg
repeat  4  [fd  50  rt  90]
pu
rt  45  fd  10
pd
setsh  29
shade
```

shape Reporter/No inputs

Reports the turtle's shape number. If more than one turtle is active, **shape** reports the shape number of the first turtle in the list reported by **who.**
See **setsh**.

Examples:

```
rg
setsh  11
show  shape
11
setsh  shape  +  1
```

shapes Command/No inputs

Saves the page that is showing, then displays the Shapes page. The Shapes page is a special page on which all of the turtle shapes are displayed and where shapes can be changed and new shapes created.

Example:

```
shapes
```

Press **Escape** to go back to your page.

show *word/list*
(**show** *word/list1 word/list2...*) Command/1 input
If inside parentheses, one or many inputs.

Prints the word or list in the command center. A list is printed with its outer brackets showing.
See **type**, **print**, **insert**, and **label**.

Examples:

```
ct
show "hello
hello
show [hello there]
[hello there]
show 22 / 7
3.1429
```

shownames Command/No inputs

Prints all variables in memory and their values, in the command center. Variables are created with **make** and **name**.
See **printnames**, **make**, and **name**.

Example:

```
name "Bob "friend
make "age 10
shownames
:friend is "Bob
:age is 10
```

sin *degrees* Reporter/1 input

Stands for **sin**e. Reports the sine of *degrees*.

Examples:

```
rg
show sin 45
0.7071
seth 45
show sin heading
0.7071
```

slowturtle Command/No inputs

LogoWriter 2.0 for IBM computers only.

Slows the turtle's drawing speed to make its movements easier to see. When LogoWriter starts up, the turtle speed is set to **fastturtle**.

sol Command/No inputs

Stands for **s**tart **o**f **l**ine. Moves the cursor on the page to the beginning of the line.

Example:

```
ct
insert [hello there]
hello there
sol
hello there
```

Reports the amount of free workspace, as a percentage of the total. You should run the **recycle** command before using **space** to find the true amount of available workspace.

Note: LogoWriter 2.0 for IBM computers reports the number of bytes of memory available, not the percentage of free memory. This is to reflect the possible differences in space available on different IBM computers, with different amounts of available memory.
See **recycle**.

Example:

```
recycle
show  space
80
```

In this example, 80% of your memory is free, and 20% is occupied by procedures, variables, etc.

LogoWriter 2.0 only.

Reports the square root of *number*. *Number* must not be negative.

Examples:

```
show sqrt 25
5
show sqrt 2
1.4142
```

The following example uses **sqrt** to create a **distance** procedure, which reports the distance between the turtle and any position on the screen.

```
to distance :position
output sqrt
   (sq (first pos) - (first :position))
 + (sq (last pos) - (last :position))
end

to sq :num
output :num * :num
end
```

In the command center, type:

```
rg
show distance [0 50]
50
show distance [50 50]
70.7106
```

Distance can report the distance between one turtle and another.

```
show distance ask 2 [pos]
56.5686
```

sspace — Command/No inputs

Stands for single **space**. Sets the printer to print single spaced text when **printtext** or **printtext80** is used. The setting will remain until a **dspace** command sets it to print double spaced text, or until the computer is turned off. See **dspace**, **printtext**, and **printtext80**.

Example:

```
repeat 10 [print [one two three]]
sspace
printtext
dspace
printtext
```

st — Command/No inputs

Stands for **s**how **t**urtle. Makes the turtle or turtles visible. See **ht**.

Example:

```
rg
tell [1 2 3]
st
ht
```

stamp Command/No inputs

Stamps a copy of the turtle shape(s) on the page. They stamp in their current color and pen state (**pu**, **pd**, **pe**, **px**).

Examples:

```
rg
stamp
fd 20
setsh 21
repeat 10 [stamp fd 20]
ht
```

startup Special Word
Procedure name or page name

A special procedure name or page name built into LogoWriter. If a page contains a procedure named **startup**, that procedure will run when you get that page. If there is a page named Startup on a LogoWriter disk, this page will automatically be displayed when LogoWriter starts up. For more information, see Appendix 2, *Startup*.

Example:

```
to startup
repeat 4 [fd 50 rt 90]
end
```

In the command center, type:

```
namepage "square
newpage
getpage "square
```

stop — Command/No inputs

Stops the procedure that is running. If the procedure that is stopped is a subprocedure, control returns to the superprocedure; otherwise, control returns to the command center. **Stop** can only be used inside a procedure.

Example:

```
to  countdown  :number
if  :number  =  0  [stop]
pr  :number
countdown  :number  -  1
end

countdown  10
```

stopall — Command/No inputs

Stops *all* running procedures and returns control directly to the command center.

Example:

```
when  "x  [stopall]
repeat  1000  [fd  1]
```

Press **Control - X** **Ctrl - X**.

tab Command/No inputs

Moves the cursor to the next tab stop.

Examples:

```
ct
insert  "Alice  tab
insert  "Bob  tab  tab
print  "Carol
Alice           Bob         Carol
```

tc Reporter/No inputs

LogoWriter 2.0 for IBM computers only.

Stands for **t**ext **c**olor. Reports the color of the text under the cursor.
See **settc**.

Note: When the cursor moves, it takes on the color of the text underneath.

tell *turtle*
tell *turtlelist* Command/1 input

Sets the turtle or turtles that will follow commands. The numbers which refer to the four turtles are 0, 1, 2, and 3.
See **ask**, **all**, and **who**.

Examples:

```
rg
tell  [0  1  2  3]
st
fd  50
tell  [0  1]
rt  90  fd  50
show  who
[0  1]
tell  3
lt  90
fd  50
```

textlen — Reporter/No inputs

Stands for **text len**gth. Reports the length, in characters, of the text on the side of the page that is showing. Any characters typed while in the label mode or with the **label** command are part of the graphics and are not counted.
See **settextpos**.

Example:

```
ct
insert  "hello
hello□
show  textlen
5
```

textpos — Reporter/No inputs

Stands for **text pos**ition. Reports a number representing the position of the text cursor in the page, in characters, from the top of the page.
See **settextpos**.

Example:

```
ct
pr  [hello  there]
hello  there
□
cu
repeat  4  [cf]
hell◎  there
show  textpos
4
```

thing *name* — Reporter/1 input

Reports the value of the variable *name*. Variables are created with the **make** and **name** commands. See **make** and **name**.

Examples:

```
make "class [Bob Alice Fred]
show thing "class
[Bob Alice Fred]
show :class
[Bob Alice Fred]
make "Fred [a very nice guy]
show thing "Fred
[a very nice guy]
show :Fred
[a very nice guy]
show thing last :class
[a very nice guy]
```

to — Special Word

Indicates the beginning of a procedure definition. **To** is used on the flip side of the page where procedure definitions are written.

Example:

```
to square
repeat 4 [fd 50 rt 90]
end
```

Produces a sound with a pitch of *frequency* which lasts for the specified amount of *time*. *Time* is measured in approximately 1/20ths of a second.
See **tone?**

Examples:

```
tone  440  30
tone  5222  30
tone  440  120
```

Note	Frequency, by Octave				
B	62	123	247	494	988
A#	58	117	233	466	932
A	55	110	220	440	880
G#	52	104	208	415	830
G	49	98	196	392	784
F#	46	92	185	370	740
F	44	87	175	349	698
E	41	82	165	330	659
D#	39	78	156	311	622
D	37	73	147	294	587
C#	35	69	139	277	554
C	33	65	131	262	523
					middle C

Not in LogoWriter 2.0.

Reports **true** if a tone is playing, or if there is a note left in the buffer of notes. If the buffer is empty it reports **false**.

Note: On LogoWriter for the Apple and the COMMODORE 64, a buffer is not used for note-playing. On these versions, **tone?** always reports **false**. See **tone** and Section 3, *LogoWriter Versions*.

Examples (not for Apple or COMMODORE computers):

```
repeat 10 [tone 440 20]
show tone?
```

Press Return Enter while a tone is playing.

```
true
tone 440 10
show tone?
```

Press Return Enter after the tone has stopped playing.

```
false
```

toollist Reporter/No inputs

Reports a list of the names of the tool procedures which are in memory. Tool procedures are procedures which have been loaded into the workspace with **gettools**. They are not shown on the flip side of the page. See **gettools**, **cleartools** and Appendix 3, *Tools*.

Example:

```
gettools "tulips
show toollist
cleartools
```

Moves the cursor on the page to the top of the text on the page.

Example:

```
ct
print [hello there]
hello there
□
print [how are you?]
how are you?
□
top
hello there
how are you?
```

towards *pos* Reporter/1 input

LogoWriter 2.0 only.

Reports the heading which points to the position given as input. **Towards** can be used to point the turtle to any position on the screen, or to another turtle.

Examples:

```
rg
seth towards [50 50]
```

The turtle turns to face position [50 50] which is in the top right section of the page.

```
ask 1 [st]
seth towards ask 1 [pos]
```

The turtle turns to face turtle 1.

```
tell all st
each [seth towards ask 1 [pos]]
```

All of the turtles turn to face turtle 1.

Used as input to **if**, **ifelse**, **and**, **or**, and **not**.
See **false**.

Examples:

```
if "true [pr "hello]
hello
ifelse and "true "false [pr "hello]
[pr "bye]
bye
```

---

type *word/list*
(**type** *word/list1 word/list2...*) Command/1 input
If inside parentheses, one or many inputs.

Prints the word or list in the command center, without a carriage return following (the cursor does not move to the next line).
See **show**, **print**, and **insert**.

Example:

```
type [show "hello]
show "hello□
hello
```

Press **Return** **Enter**.

U

undo — Command/No inputs

Not in LogoWriter 2.0 for IBM computers.

Brings back any text that was cleared by a **ct** or **cp** command. In order to recover text that was accidentally lost, **undo** must be used before the contents of LogoWriter's temporary text buffer are changed; i.e., before any instruction such as **print**, **insert**, or **delete** changes the text on the screen, and before you save the page.

Example:

```
ct
repeat 5 [print [hello]]
ct
undo
```

unlock — Command/No inputs

LogoWriter 2.0 only.

Removes the protection from a locked page. Use **unlock** if you want to make changes to a locked page, or to erase a locked page.
See **lock**.

Undoes the effect of **select**. Any characters that have been selected will no longer be highlighted. It can also be used to undo the highlighting effect of **search**.
See **select** and **search**.

Examples:

```
ct
insert [hello there]
hello there□
select
sol
hello there
unselect
hello there
top
search "there
hello there
unselect
hello there□
```

wait *number* Command/1 input

Causes a pause for *number* 20ths of a second. An input of 10 causes a pause of approximately half a second.

Example:

```
repeat 90 [fd 1]
repeat 90 [fd 1 wait 2]
```

when *letter listtorun* Command/2 inputs

Programs a control-key event. Whenever the **control** key and the *letter* key are pressed, *listtorun* is run. *Listtorun* will be run immediately when the key combination is pressed, regardless of whether another program is running. Only ten keys can be used as **control** keys: N O P Q R V W X Y Z. Control-key events are only saved on a page if the **when** instruction is put inside a procedure.

Note: If **readlist**, **readlistcc**, or **readchar** are waiting to read from the keyboard, control-keys will be ignored. See **clearevents**.

Examples:

```
when "z [pr "hello]
when "x [fd 10]
```

Press the **Control - Z** **Ctrl - Z** and **Control - X** **Ctrl - X** key combinations a few times.

```
clearevents
```

who Reporter/No inputs

Reports the number or list of numbers of the turtles that are active. If **who** is used in an instruction list run by **each**, it will report one turtle number at a time.
See **tell**, **ask**, and **each.**

Examples:

```
rg
tell [0 1 2 3]
st
fd 50
show who
[0 1 2 3]
ask [0 1][rt 90 fd 50]
show who
[0 1 2 3]
```

word *word1 word2*
(**word** *word1 word2 word3...*) Reporter/2 inputs
If inside parentheses, one or many inputs.

Combines all of its inputs into one word and reports that word.

Examples:

```
show word "super "man
superman
show (word "stick "it "together)
stickittogether
```

Reports **true** if the input is a word; otherwise, it reports **false**.

Examples:

```
show word? "hello
true
show word? [hello]
false
show word? 123
true
if word? "hello [show count "hello]
5
```

xcor Reporter/No inputs

LogoWriter 2.0 only.

Stands for **x-coor**dinate. Reports a number representing the horizontal position of the turtle on the page.
See **ycor**, **setx**, and **setpos**.

Examples:

```
setpos  [20  50]
show  xcor
20
setx  -30
show  xcor
-30
```

ycor — Reporter/No inputs

LogoWriter 2.0 only.

Stands for **y-coor**dinate. Reports a number representing the vertical position of the turtle on the page.
See **xcor**, **sety**, and **setpos**.

Examples:

```
setpos [20 50]
show ycor
50
sety -80
show ycor
-80
```

ProDOS Primitives

These primitives are only in LogoWriter 2.0 for Apple *II* computers. They have been included in that version to take advantage of the special features of the ProDOS disk operating system. For more information on how LogoWriter uses the features of ProDOS, see Section 3, *LogoWriter Versions*.

LogoWriter 2.0 uses the percent (%) character as a special delimiter for ProDOS pathnames. In other ProDOS programs the slash (/) is used.

---

bye Command/No inputs

Quits LogoWriter and returns to ProDOS.

---

copyfile *name/pathname1 name/pathname2*
Command/2 inputs

Makes a copy of a file. The original file is not affected. Any kind of ProDOS file can be copied - a LogoWriter page, the Shapes page, an ASCII file, a binary file. The first input, *name/pathname1*, is the name (or the pathname) of the original file; the second input, *name/pathname2*, is the name for the new file.

Examples:

This instruction copies a file within a directory.

```
copyfile "triangle "triangle.copy
```

This instruction copies a file from the main directory of one disk (named **adam**) to the main directory of another (named **eve**).

```
copyfile "%adam%triangle "%eve%triangle
```

This instruction copies from one subdirectory to another, on the same disk. Note that the page itself can have the same name, if it's in a different subdirectory.

```
copyfile  "%datadisk%directory1%page1
"%datadisk%directory2%page1
```

---

createdir *pathname* Command/1 input

Makes a new subdirectory. The first element of *pathname* is the volume name, the name of the disk. The last element of *pathname* is the name of the new subdirectory.

Subdirectories may contain subdirectories. **Createdir** can create a new subdirectory at any level as long as *pathname* specifies the volume, and the path of subdirectories which will contain the new subdirectory.

Note: ProDOS limits the length of a pathname to 64 characters. This length includes the % characters.

Examples:

The following instruction creates a new subdirectory named **karen**.

```
createdir  "karen
show  directories
[KAREN]
```

The following instruction creates a new subdirectory **work** within the subdirectory **karen**.

```
createdir  "karen%work
```

Note: To erase a subdirectory, use **erasefile** or erase from the Contents page with the **erase to end of line** keys. The subdirectory must be empty before it can be erased.

directories Reporter/No inputs

Reports a list of subdirectory names. If you are using the main directory, then **directories** will report the subdirectories listed in the main Contents page. If you are in a subdirectory, **directories** will report the names of the subdirectories at that level. You can change directories from the Contents page, or use the **setprefix** command. To return to the main directory, use **setprefix** with the volume name alone as input, or use **setdisk**.
See **setprefix**, **prefix**, and **setdisk**.

Examples:

```
setprefix  "%datadisk
show  directories
[KAREN  FRED  BILL]
setprefix  "karen
show  directories
[PROJECT1  PROJECT2]
```

erasefile *name/pathname* Command/1 input

Erases a file from the disk. Unlike **erpage**, **erasefile** can erase any kind of file, not just LogoWriter pages. **Erasefile** is the only command that can erase the Shapes page.

Erasefile can also erase a subdirectory if the subdirectory is empty. To find out what files are in the current subdirectory, use **filelist**.

Examples:

The following instruction erases a file named **bear**, which is in the current directory.

```
erasefile  "bear
```

If you use the full pathname, **erasefile** can erase a file from a different subdirectory, or even a different disk. In the following instruction, the disk (volume) name is **datadisk**, the subdirectory name is **karen**, and the file to be erased is **triangles**.

```
erasefile  "%datadisk%karen%triangles
```

---

filelist Reporter/No inputs

Reports a list of all files in the current directory. The file name is followed by a code representing the file type, in this form: [filename filetype]. The file type codes are as follows:

lwp	LogoWriter page
lws	LogoWriter system file
bin	binary graphic file
dir	directory
txt	ASCII text file
prn	LogoWriter printer file
prc	LogoWriter printer card file
sys	ProDOS system file

Examples:

```
show  online
[%LOGOWR%  %WORK%]
setprefix  "%work
show  filelist
[[PRODOS  sys][SHAPES  lws][BARCHART
lwp][PIECHART  lwp]  [GRAPHICS  dir]
[CHARTS  dir]]
show  directories
[GRAPHICS  CHARTS]
setprefix  "%work%charts
show  filelist
[[LINECHARTS  lwp][PIE1  lwp][PIE2
lwp][SHAPES  lws]]
```

Loads a graphic (binary) file onto the current page. The graphic image is overlaid onto any graphics already on the page.

Files created by other programs can be loaded by **loadpic**, as long as they are standard binary files. The bottom portion of a full screen graphic (where the command center is) will not be displayed.

Any LogoWriter graphic can be saved as a binary file with **savepic**.

Note: Loadpic can load any binary file onto the page, but the results of loading a non-graphic file may look strange.
See **savepic** and **filelist**.

Examples:

```
rg ct
repeat 120 [fd 50 bk 50 rt 1]
savepic "pic1
clean setc 2
repeat 120 [fd 50 bk 50 rt 1]
savepic "pic2
clean setc 3
repeat 120 [fd 50 bk 50 rt 1]
savepic "pic3
clean ht
loadpic "pic1
loadpic "pic3
loadpic "pic2
```

loadtext *name/pathname* Command/1 input

Loads an ASCII (text) file onto the current page. The text is added at the bottom of whichever side of the page is showing.

Files created by other programs, which have been saved as text only (ASCII) files, can be loaded with **loadtext**.

The text from either side of a LogoWriter page can be saved as an ASCII file with **savetext**.

Example:

```
ct
repeat 4 [print [A few words]]
savetext "LogoWritertext
show filelist
[[LOGOWRITERTEXT txt]]
ct
loadtext "logowritertext
```

The text file can now be loaded into a program such as the AppleWorks™ word processor with the "Create a new file from an ASCII file" option.

online Reporter/No inputs

Reports a list of the volume names (that is, disk names) that are currently available (inserted in disk drives or on a network). **Online** is useful when you insert a disk but don't know its volume name. The volume name is the first part of a pathname.
See **setprefix**, **prefix**, and **setdisk**.

Example:

```
setprefix  "%scrapbook
Directory  not  found
show  online
[%CLASSDISK%  %DATADISK%]
setprefix  "%datadisk
show  prefix
%DATADISK%
setdisk  "a
show  prefix
%CLASSDISK%
```

---

prefix Reporter/No inputs

Reports the current prefix setting that was set by **setprefix** or **setdisk**. The name reported by **prefix** is the first part of a pathname, with the page name or file name missing.
See **setprefix**.

Examples:

The current disk's volume name is **logoprograms**.

```
show  prefix
%LOGOPROGRAMS%
```

The following instruction uses **setprefix** to set the subdirectory to **brian**.

```
setprefix  "%logoprograms%brian%
show  prefix
%LOGOPROGRAMS%BRIAN
```

Use the volume name alone as input to **setprefix** to set the current directory back to the main directory of the disk.

```
setprefix  "%logoprograms
```

Setdisk also changes the prefix to the main directory of a disk.

```
setdisk  "b
show  prefix
%SCRAPBOOK%
```

rename *name/pathname1 name/pathname2*
Command/2 inputs

Changes the name of a file. If the new name is already used in the directory, a message will be printed. **Rename** can only change a file's *name*; it can't copy the file from one directory to another. A locked file can't be renamed. See **copyfile**, **lock**, and **unlock**.

Example:

```
show pagelist
[SQUARES TRIANGLES]
rename "squares "newsquares
show pagelist
[NEWSQUARES TRIANGLES]
```

savepic *name/pathname* Command/1 input

Saves the graphics part of the page as a standard ProDOS graphic (BIN or binary) file. The graphics file can then be loaded back onto the page by **loadpic**. Other programs which use standard graphic files can also load the file.

Files saved with **savepic** are not LogoWriter pages, and do not appear in Contents. To find the names of every file in the directory, use **filelist.**

Standard graphic files are much larger than LogoWriter pages. A BIN file saved with **savepic** consumes 17 blocks on a disk; whereas a LogoWriter page with the same graphic might consume from 4 to 7 blocks. It is recommended that you keep graphic files on a separate disk from LogoWriter pages, or you will quickly run out of disk space.
See **loadpic**, **savetext**, and **filelist**.

savetext *name/pathname* Command/1 input

Saves the text on the current side of the page as a standard text (ASCII) file. The text can be loaded back with **loadtext**. Text files can be accessed by other word processors or programs that use ProDOS text files.

Files saved with **savetext** are not LogoWriter pages, and do not appear in Contents. To find the names of every file in the directory, use **filelist.**

Example:

On the flip side of the page, type some procedures.

With the flip side showing, go to the command center, and type:

```
savetext  "proc.text
```

The procedures are saved into a text (ASCII) file named "proc.text." The procedures can be loaded into another program, such as *LCSI LOGO II*™. Files created by *LCSI LOGO II* can be loaded into LogoWriter 2.0 for Apple *II* computers with **loadtext**.

setprefix *pathname* Command/1 input

Changes the current volume and/or subdirectory name to *pathname*.

Note: Once **setprefix** is used, LogoWriter will save pages on the named disk and directory. If necessary, LogoWriter will search every disk drive and device, trying to find the volume and/or directory name that matches the prefix setting.

To return to the main directory of a disk, use the volume name alone as input to **setprefix**, or use **setdisk**. See **setdisk**, **prefix**, **online**, and **directories**.

Examples:

```
show  online
[%DATADISK%]
show  directories
[DIR1  DIR1  DIR3]
setprefix  "%datadisk%dir3
show  prefix
%DATADISK%DIR3%
setprefix  "%datadisk
pr  prefix
%DATADISK%
```

---

setslot *number* Command/1 input

Sets LogoWriter to save and load to the disk drive plugged into slot *number*. The slot *number* can be any number from 2 through 6. This is designed for network users who want to redirect saving and loading from a network to an attached floppy disk. It is also useful for Apple IIGS users who want to switch between the 3 1/2" disks (usually in slot 5) and 5 1/4" disks (usually in slot 6).

---

slot Reporter/No inputs

Reports the disk drive slot number to which saving and loading is directed. Normally this will report the number of the disk drive slot from which LogoWriter started up. See **setslot** and **setdisk**.

IBM DOS Primitives

These primitives are only in LogoWriter 2.0 for IBM computers. They have been included in that version to take advantage of the special features of the IBM DOS operating system. For more information on how LogoWriter uses the features of DOS, see Section 3, *LogoWriter Versions*.

---

chdir *name/pathname* Command/1 input

Stands for **ch**ange **dir**ectory. Changes the directory for saving and loading. To change to a directory within the current directory, you can use a directory name without specifying the path (i.e. without any backslash characters). To change to a directory in a different path, use a pathname containing backslashes to separate the individual names. Special DOS characters, the backslash (\) and two periods (..), can be used as input to **chdir**.

Changing the disk drive for saving and loading (with **setdisk**) also changes the directory.
See **currentdir**, **directories**, **mkdir**, **setdisk**, and **disk**.

Examples:

```
mkdir  "dir1
show  directories
[DIR1]
chdir  "dir1
show  currentdir
\dir1
mkdir  "subdir1
show  directories
[SUBDIR1]
chdir  "subdir1
show  currentdir
\dir1\subdir1
```

Special DOS characters can be used as input to **chdir**. Two periods (**..**) "back up" one level of directory (as does pressing **Escape** at the Contents page). The backslash (\) goes to the root directory.

```
show  currentdir
\dir1\subdir1
chdir  ".. 
show  currentdir
\dir1
chdir  "\

show  currentdir
\
show  directories
[DIR1]
```

Two periods back up one level.

The backslash goes to the root directory.

When **currentdir** reports the backslash (\), the current directory is the root.

---

copyfile *name/pathname1 or list of names*
name/pathname2 Command/2 inputs

Makes a copy of a file. The original file is not affected. Any kind of file can be copied - a LogoWriter page, the Shapes page, an ASCII file, a binary file. The first input is the name (or the pathname) of the original file; the second input, *name/pathname2,* is the name for the new file. The entire filename must be used, including the extension characters (see the filename extensions in the listing for **filelist**). If the new file does not end with the extension characters **lwr**, it will not be a LogoWriter page listed on Contents.

The first input can be a list of file names. In this case, the files will be merged (appended) into one file. This method of merging files works only with text files and binary program files, and does not work with files that have special formats, such as LogoWriter pages or picture files.

Examples:

This instruction copies a page.

```
copyfile  "triangle.lwr  "triangle2.lwr
```

This instruction copies a page named **john** from the root directory of a disk to a subdirectory named **subdir1**, which is in the subdirectory **dir1**.

```
copyfile  "\john  "\dir1\subdir1\john
```

---

currentdir Reporter/No inputs

Reports the name of the current directory; i.e. the directory containing the current page. If the current directory is the root directory (which has no name), **currentdir** reports the backslash.
See **directories**, **chdir**, **mkdir**, and **setdisk**.

Examples:

```
show  currentdir
\
show  directories
[DIR1]
chdir  first  directories
show  currentdir
\dir1
make  "firstlevel  currentdir
chdir  "\
show  currentdir
\
chdir  :firstlevel
show  currentdir
\dir1
```

---

directories Reporter/No inputs

Reports a list of the subdirectories contained in the current directory. These directories are listed in the Contents page of the current directory.

dos Command/No inputs

Saves the page and leaves LogoWriter, returning to the operating system.

erasefile *name/pathname* Command/1 input

Erases a file from the disk. Unlike **erpage**, **erasefile** can erase any kind of file, not just LogoWriter pages. **Erasefile** is the only command that can erase the Shapes page. To erase a file that is not in the current directory, use a pathname comprising the name of each directory in the path, and finishing with the name of the file itself. Each name in the pathname must be separated by a backslash (\). You must use a complete file name, including the filename extension.
See **filelist** for the table of filename extensions.

Examples:

The following command erases a LogoWriter page named **bear** from the current directory:

```
erasefile  "bear.lwr
```

The following instruction erases a LogoWriter page named **rocket** from a directory named **dir1**:

```
erasefile  "dir1\rocket.lwr
```

The following instruction erases the Shapes page from the current directory:

```
erasefile  "shapes.shp
```

The following instruction erases the Shapes page from the directory **dir1**:

```
erasefile  "dir1\shapes.shp
```

filelist Reporter/No inputs

Reports an alphabetically ordered list of all files in the current directory. Unlike **pagelist**, which reports only the names of LogoWriter pages, **filelist** reports all files including their extensions. LogoWriter files use the following two extensions:

LogoWriter page	.LWR
Shapes page	.SHP

A text file saved with **savetext** or **savepic** can have any extension. Extension characters which have special meaning to DOS are as follows:

Assembly source file	.ASM
Batch file	.BAT
Executable file	.COM
Executable file	.EXE
Library file	.LIB
Linker list file	.MAP
Object file	.OBJ
System file	.SYS

Example:
The current directory contains a LogoWriter page named **story**, a Shapes page, and a file named **data.txt**.

```
show  filelist
[DATA.TXT  SHAPES.SHP  STORY.LWR]
```

loadpic *name/pathname* Command/1 input

Loads a graphic file, or the graphics from a LogoWriter page, onto the current page. The graphic image is overlaid onto any graphics already on the page.

Graphics files are created with **savepic**.

See **savepic**, **loadtext**, **savetext**, and **filelist**.

Example:

```
cg
loadpic  "page1.lwr
loadpic  "page2.lwr
```

loadtext *name/pathname* Command/1 input

Loads a text file onto the side of the page that is facing. The text is added to the end of any text that is already on the page. The file can be a file which was saved with **savetext**, or almost any other file.

If the file is a LogoWriter page, only the front side of the page is loaded.

If the file is not a text file (if it contains graphics or other binary data), LogoWriter will load it nonetheless, even though not all of the characters can be displayed, and those that can be displayed may not make sense as text.

Loadtext and **savetext** can be used to access text files from other word processors, and can even be used to edit or create DOS files, such as .BAT files. See also **savetext** and **load**.

mkdir *name/pathname* Command/1 input

Stands for **mak**e **dir**ectory. Creates a new directory. If the input contains no backslash (\) characters, the directory will be created in the current directory, and will be listed in the current directory's Contents page. If the full pathname is used, the new directory will be created in the specified location.

Examples:

The following example makes a directory named **dir1** within the current directory:

```
mkdir  "dir1
show  directories
[DIR1]
```

The following instruction makes a new directory **dir2**, in the directory **dir1**. Since that directory is two levels below the current directory, its name will not be listed by **directories**.

```
mkdir  "\dir1\dir2
show  directories
[DIR1]
```

To access the new directories, use **chdir**, or press **Escape** to go to the Contents page. **Dir1** will be listed there. Put the cursor on it and press **Enter**. The **dir1** Contents page will list **dir2** as a subdirectory. Enter that directory by putting the cursor on it and pressing **Enter**.

---

rename *name/pathname1 name/pathname2*
Command/2 inputs

Changes the name of a file. The first input is the name (or the pathname) of the existing file; the second input is the new name for the file. The entire filename must be used, including the extension characters (see the filename extensions in the listing for **filelist**).

Example:

```
rename  "triangle.lwr  "triangle2.lwr
```

---

rmdir *name/pathname* Command/1 input

Stands for **r**emove **dir**ectory. Erases a directory. The directory must be empty of files before it can be erased. To list the files in a directory, enter the directory (by choosing its name from the Contents page, or by using **chdir**), and use **filelist** to list the files. **Erasefile** can be used to erase the files in the directory, before the directory can be erased with **rmdir**. You cannot erase the current directory.

Examples:

```
show  currentdir
\
show  directories
[DIR1  PROJECTS]
rmdir  "dir1
show  directories
[PROJECTS]
chdir  "projects
show  filelist
[SHAPES.SHP]
erasefile  "shapes.shp
chdir  "\
rmdir  "projects
```

The directory **dir1** has been erased.

The only file in the directory is the Shapes page.

---

savepic *name/pathname* Command/1 input

Saves the graphics on the page in a special graphics-only file. The graphic file can then be loaded back onto any page with **loadpic**. Files saved with **savepic** are not LogoWriter pages and do not appear in Contents. To find the names of every file, use **filelist**.

Example:

```
repeat  5  [forward  50  right  144]
savepic  "star.gfx
cg
loadpic  "star.gfx
```

---

savetext *name/pathname* Command/1 input

Saves the text on the facing side of the page in a text file. The text file is not a LogoWriter page, and does not show on the Contents page. Any filename extension can be used as part of the file name (although you should follow DOS conventions for extension characters). Files saved by **savetext** can be accessed by other programs which can read standard text files. **Savetext** can be used to save .BAT and .COM files.
See **loadtext**, **loadpic**, **savepic**, and **load**.

System Primitives

These are special primitives which can affect the computer's memory, and the LogoWriter system itself. They let you directly address memory locations - and for that reason they can be dangerous, since you can destroy the contents of your workspace or cause LogoWriter to crash. Make sure your work is saved before you use these primitives.

System Primitives for Apple II Computers

To use these primitives, you must know the areas of memory in which data or programs can be put without risk. In a 128K Apple *II*e or *II*c, the free block of memory is between address 39680 (9B00 Hex) and 40714 (9F00 Hex).

In an Apple IIGS there is more than 128K available. LogoWriter only uses 128K (memory banks 0 and 1). You are free to use memory above bank 1 (above address 131072).

.bload *name/pathname address* Command/2 inputs

Stands for **b**inary **load**. Loads a binary-format file, which may consist of data or machine-language code, into an area of memory starting at *address*. *Address* must be expressed in decimal form.

.bsave *name/pathname address length*

Command/3 inputs

Stands for **b**inary **save**. Copies an area of computer memory into a binary file. The memory region starts at *address* and is *length* bytes long. Both *address* and *length* must be in decimal form.

.call *address* Command/1input

The **.call** command transfers control to the machine language subroutine which starts at *address*. *Address* must be in decimal form. An RTS instruction in the subroutine returns control back to the LogoWriter instruction following the **.call** command. Normally the program will have been loaded into memory with **.bload**.

.deposit *address byte* Command/2 inputs

Writes *byte* into the memory location *address*. Both *byte* and *address* must be in decimal form.

.examine *address* Reporter/1 input

Reports the contents of the memory location *address*. The value reported will be a decimal number. *Address* must be a decimal number.

System Primitives for IBM Computers

These primitives are used to directly access the computer's memory and hardware. They are intended for use by software developers and our internal technical support department to customize LogoWriter for special input and output devices.

The names of these primitives start with a period. This is a warning that they are potentially dangerous: by writing to important locations in the computer's memory you can crash LogoWriter.

.deposit *address word* Command/2 inputs

Writes *word* into the memory location *address*. Both *word* and *address* must be in decimal form. *Word* is any decimal value from 0 to 65535 (FFFF hex). *Address* is any absolute memory location to 1 megabyte. There are no "guaranteed safe" memory locations in LogoWriter, but the most recommended area to use for reading and writing is the user interrupt area.

.examine *address* Reporter/1 input

Reports the contents of memory location *address*, as a decimal number from 0 to 65535 (FFFF hex). *Address* is any absolute memory location to 1 megabyte, expressed as a decimal number.
See **.deposit**, **.in**, and **.out**.

.in *address* Reporter/1 input

Reads from an input-output port which is located at the specified address, and reports the value found there. The ports are the devices available, such as serial lines, paddle connectors, keyboard, etc. The value that is reported will be a byte, in decimal form. To find the address of the ports on your computer, refer to the technical information available from the manufacturer.
See **.examine**, **.deposit**, and **.out**.

.out *address byte* Command/2 inputs

Sends a byte of information to the input-output port located at the specified address. The ports are the devices available, such as serial lines, paddle connectors, keyboard, etc. *Address* and *byte* must be decimal values: address can be any number up to 65535 (FFFF hex). To find the address of the ports on your computer, refer to the technical information available from the manufacturer.
See **.examine**, **.deposit,** and **.in**.

.version Reporter/No inputs

Reports the version number of the LogoWriter program and the current graphics and printer setting. Check **.version** if you suspect that the printer or graphics setting is incorrect. Refer to the version number if you call or write for technical assistance.

Example:

```
show  .version
```

Section 3: LogoWriter Versions

LogoWriter runs on the Apple *II*, IBM PC, and COMMODORE 64 computers in essentially the same way. There are minor differences due to computer keyboards, graphics, and memory. For Apple *II* and IBM PC computers there are two versions of LogoWriter available: LogoWriter 1.1 and LogoWriter 2.0.

This section details the differences between each of the versions of LogoWriter.

Chapter I, *LogoWriter 2.0 for Apple II Computers*, describes the keys and colors used by this version. As well, there is a section explaining how LogoWriter uses the special features of the ProDOS operating system.

Chapter II, *LogoWriter 2.0 for IBM Computers*, describes the keys used and the colors available with the various graphic adapters that LogoWriter can use. There is also a section explaining how LogoWriter takes advantage of the special features of the IBM DOS operating system.

Chapter III, *LogoWriter 1.1*, describes the specific details of each currently available version of LogoWriter 1.1: LogoWriter 1.1 for 64K Apple *II* computers including the *II+*, LogoWriter 1.1 for the IBM PC and the IBM PC*jr*, and LogoWriter for the COMMODORE 64.

I LogoWriter 2.0 for Apple II Computers

LogoWriter 2.0 for Apple *II* computers requires 128K of memory to run. It offers approximately double the page size and more than triple the workspace as version 1.1, and utilizes the ProDOS disk operating system. It runs on 3 1/2" and the 5 1/4" disks, as well as on ProDOS hard disks and ProDOS based classroom networks.

LogoWriter 2.0 contains extra primitives which are not included in version 1.1. Many of these are in the alphabetical listing of primitives.

Other new primitives have been added to version 2.0 in order to take advantage of the features of ProDOS. These primitives are listed in the chapter, *ProDOS Primitives*. An overview of ProDOS concepts and ProDOS LogoWriter features follows at the end of this chapter.

The ProDOS primitives should be considered optional: you can use LogoWriter 2.0 in the same way as you use version 1.1, and you will have the advantages of larger pages and more workspace. The extra primitives support classroom networks, hard disks, and ProDOS based file functions.

LogoWriter Keys

Apple - Combination Keys

To use any of these keys, hold down one of the [Apple] keys and press the key.

[Apple] - 1	Select
[Apple] - 2	Cut
[Apple] - 3	Copy
[Apple] - 4	Paste
[Apple] - 6	Erase to End of Line
[Apple] - 8	Label
[Apple] - 9	Turtle-Move
[Apple] - 0	Help
[Apple] - U	Up
[Apple] - D	Down
[Apple] - F	Flip
[Apple] - S	Stop
[Apple] - →	Next Screen
[Apple] - ←	Previous Screen
[Apple] - ↑	Top of Page
[Apple] - ↓	Bottom of Page
[Apple] - B	Beginning of Line
[Apple] - E	End of Line
[Apple] - O	Open a New Line

Apple LogoWriter Colors

The Apple *II*'s color numbers are 0 through 5, but you can use any numbers. Numbers larger than 5 "wrap": **setc 6** is interpreted as **setc 0**, **setc 7** as **setc 1**, and so on.

Number	Color
0	Black
1	White
2	Green
3	Violet
4	Orange
5	Blue

Game Paddle Inputs

LogoWriter for Apple *II* computers accepts a pair of game paddles or a joystick plugged into the game port socket. The primitive **paddle** reads the position of a game controller, and reports it as a number from 0 through 255.

On game paddles, the fully counterclockwise position reports 0; the fully clockwise position reports 255.

Paddle 0 reads the paddle labeled paddle 0.

Paddle 1 reads the paddle labeled paddle 1.

One joystick can be used in place of two game paddles. **Paddle 0** reads the up-and-down movement of the joystick. All the way down reports 0; all the way up reports 255.

Paddle 1 reads the side-to-side movement of the joystick. All the way to the left reports 0; all the way to the right reports 255.

Note: Some brands of paddles and joysticks have a restricted range - not all the way down to 0 or up to 255.

The ProDOS Features of LogoWriter 2.0

ProDOS is a system for saving, loading, and organizing files on a disk. It is a standard system used by most 128K Apple *II* family computers. It is *not* the system used by LogoWriter 1.0 and 1.1 for Apple computers.

This is a general introduction to the ProDOS features of LogoWriter 2.0. It is not a ProDOS reference guide. For more information, refer to a complete ProDOS user's manual.

Following the introduction to ProDOS are two short tutorial examples using LogoWriter's ProDOS commands and reporters.

Files and Pages

A LogoWriter scrapbook contains pages. A "page" is a special kind of file which mixes text and graphics. LogoWriter's page commands work on LogoWriter pages, not other kinds of files. The Contents page lists pages only. **Getpage** gets pages only; **erpage** erases pages only.

A ProDOS disk can contain different kinds of files: graphics files, text files, system files, and others. The ProDOS primitives are specifically intended for access to files other than LogoWriter pages; for example, to add an AppleWorks™ document to a LogoWriter page and vice versa. These primitives also allow LogoWriter to run on most classroom networks.

In general, if you are working with LogoWriter pages, use the "page" commands: **getpage**, **erpage**, **newpage**, etc. If you want access to ProDOS files, use the file commands: **copyfile**, **loadpic**, **savepic**, **loadtext**, **savetext**, and **erasefile**. The ProDOS directory features require the pathname-oriented primitives: **setprefix**, **prefix**, **createdir**, **online**, and **directories**. All of these primitives are described in the chapter, *ProDOS Primitives*, in Section 2.

Disks and Volumes: Setdisk Versus Setprefix

LogoWriter generally refers to a disk by its location - that is, LogoWriter will get pages or save pages onto a disk in drive A (drive 1), even if you swap disks. The easiest way to copy a page from one disk to another is to switch disks while a page is showing on the screen and press **Escape**. The page is saved onto the new disk.

The ProDOS system works in a different manner: a disk is given a name, called the *volume* name. The computer remembers this name, and always tries to save a file on the same disk that it came from originally, unless you've given typed-in commands to change this.

LogoWriter 2.0 gives you the best of both worlds. There are two sets of commands: one set works the "LogoWriter" way, and the other set works the "ProDOS" way.

Using Setdisk

Setdisk sets LogoWriter to use a specific disk drive for getting and saving pages. **Setdisk "a** sets LogoWriter to disk drive A (drive 1); **setdisk "b** sets LogoWriter to drive B (drive 2). After telling LogoWriter to use drive A, you can swap disks, and LogoWriter will still use drive A.

Note: When LogoWriter starts up, it checks to see whether your computer is using floppy disks. If so, it automatically uses the **setdisk** - "standard LogoWriter" - way of operating. If you're using LogoWriter on a hard disk or a classroom network, it uses the **setprefix** - "full ProDOS" - way of referring to volume and directory names for getting and saving pages.

If you are using LogoWriter in a mixed computer lab with more than one kind of computer, use **setdisk** so that each LogoWriter computer runs the same way.

Classroom network users who want to save on floppy disk drives will have to use an extra command, **setslot**, to redirect the computer from the network to the floppy disk drives.

Using Setprefix

LogoWriter 2.0 gives you the option of using "full ProDOS" to get pages and save pages on a named disk. This method is much more flexible - it allows you to organize your pages in directories, and can give you more than one Shapes page on a disk - but it is also more complex.

When you use **setprefix**, you are telling LogoWriter to always save your pages on the same disk, no matter which disk drive it may be in. If you then remove the disk from the disk drive, LogoWriter will search for the disk in every disk drive available, and if it fails to find the disk, it will beep or print an error message. The simple shortcut of swapping disks to copy a page will no longer work. To change the disk or directory name, use **setprefix** with a different disk name or directory name.

If, after experimenting with prefixes and pathnames, you prefer to use the simpler method of referring to a disk by its location, use the **setdisk** command. LogoWriter will then get and save pages on disk drive A or B.

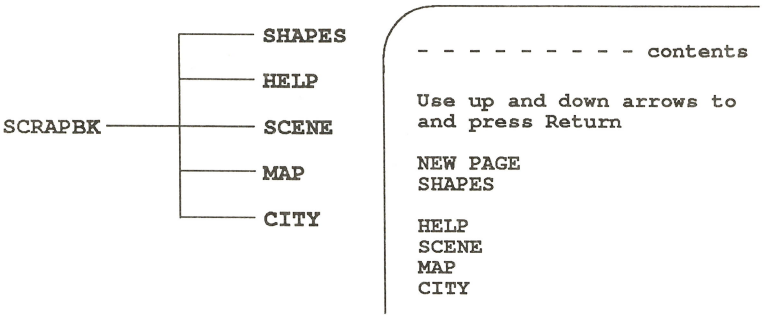
Directories

The volume (disk) can contain files (such as LogoWriter pages), and groups of files. The volume has a list of the files and groups of files it contains, called the *main directory*. The groups of files within the main directory are *subdirectories*.

When you start up LogoWriter 2.0 and begin working on pages, these pages are saved in the main directory. This is the same as in other versions of LogoWriter.

A LogoWriter disk, not using subdirectories

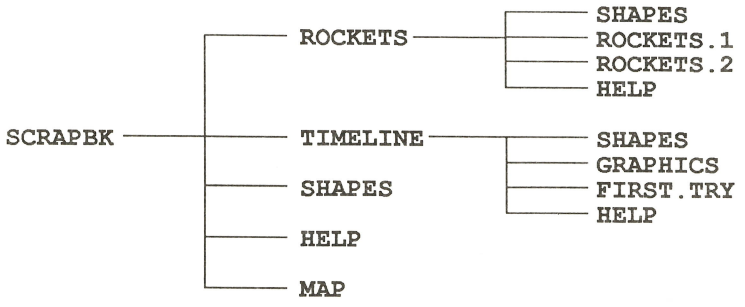
The volume (the disk) | The main directory contains the pages | The Contents page would like this:



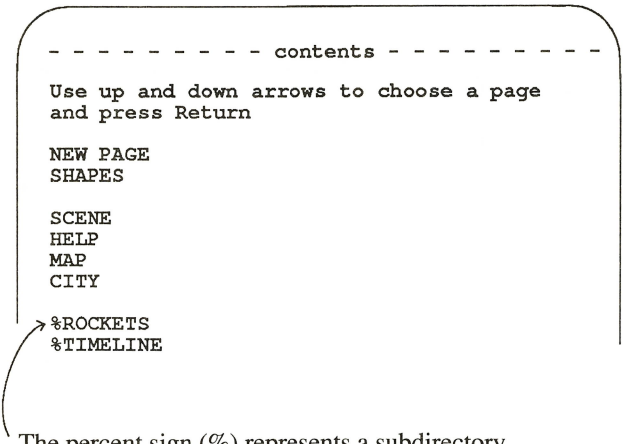
When LogoWriter uses subdirectories, each subdirectory becomes, in effect, a complete LogoWriter scrapbook. The division could look like this:

A LogoWriter disk, using subdirectories

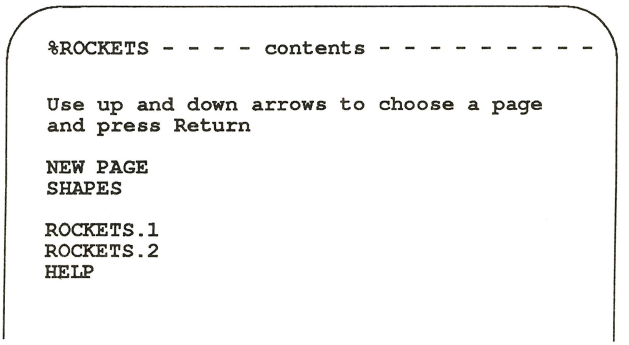
The volume | The main directory contains subdirectories and pages | Pages inside subdirectories



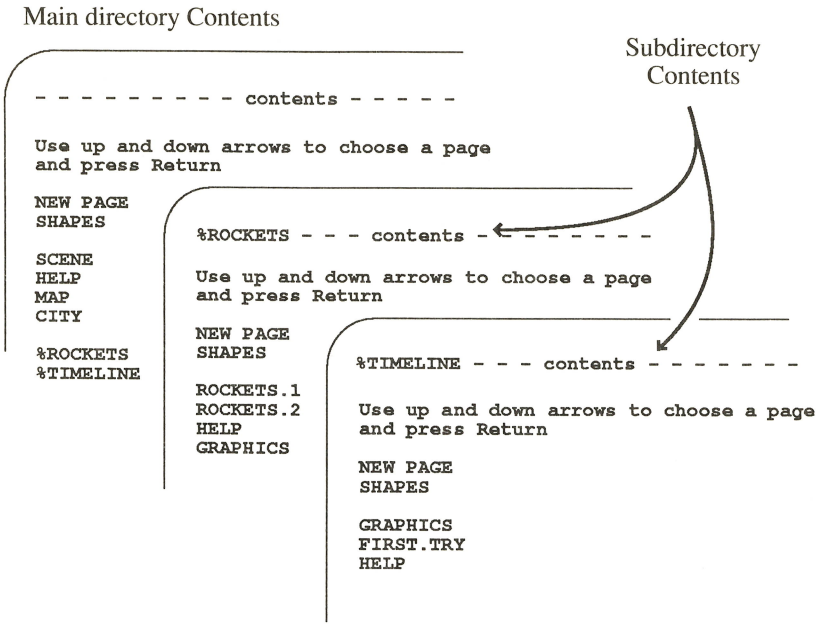
You can use a subdirectory to store a completed project on a disk, with its own Shapes page. A teacher can keep a disk of class projects, with a subdirectory for each student. Subdirectories show on Contents like pages, but with a percent sign:



The percent sign (%) represents a subdirectory.
To open a directory, put the cursor on the name and press **Return**.



To return to the main directory from a subdirectory, press **Escape**.

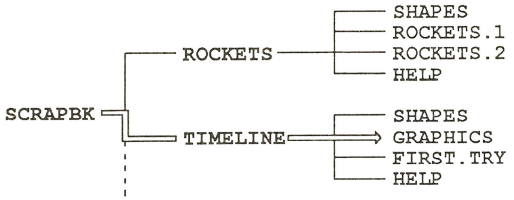


Summary

Open a LogoWriter subdirectory from the Contents page by putting the cursor on the subdirectory name and pressing **Return**. Leave a subdirectory from the subdirectory's Contents page by pressing **Escape**. To create a subdirectory or to erase one, you must use ProDOS primitives.

Pathnames

The key to taking advantage of the ProDOS features is the concept of *pathname*. A pathname represents the route (or path) that must be followed to find a file.



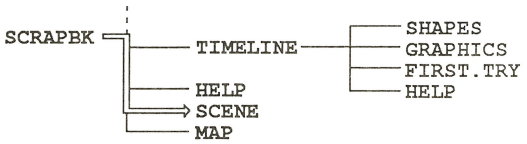
The complete pathname for the page Graphics is:

%SCRAPBK%TIMELINE%GRAPHICS

Volume name — Directory name — Page name

The percent sign is used to separate the parts of a pathname.

The pathname is one long word which indicates the path. It starts with the volume name, followed by the directory name, then the page name. Each component is separated by the percent sign. The volume name always starts *with* a percent sign; the file name must end *without* a percent sign.



The page Scene is not in a subdirectory, so its pathname is:

%SCRAPBK%SCENE

Volume name — Page name

Experienced ProDOS users will note one change from the ProDOS norm. ProDOS uses the slash character (/) to separate names within a pathname. However, LogoWriter already uses the slash as the division sign. So within LogoWriter, the slashes are replaced by percent signs. A pathname which LogoWriter calls

```
%MYVOLUME%MYDIRECTORY%MYPAGE
```

is referred to by ProDOS as:

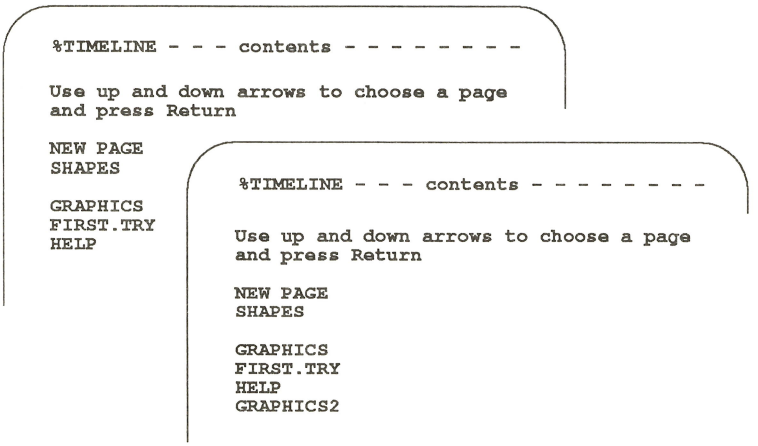
```
/MYVOLUME/MYDIRECTORY/MYPAGE
```

This conversion is within LogoWriter only. It does not affect the way the files are named or referred to in other ProDOS based programs.

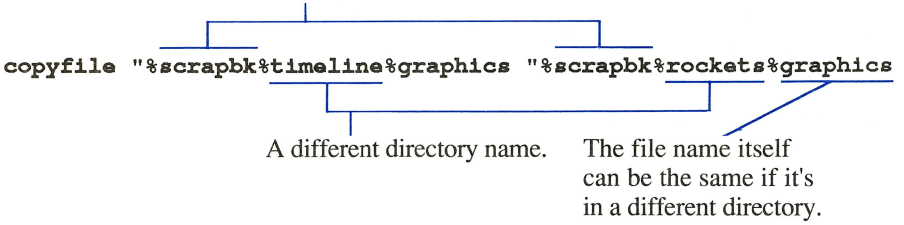
The entire pathname is not required as long as your instruction refers to files in the current directory. For example, within the subdirectory **timeline**, the instruction:

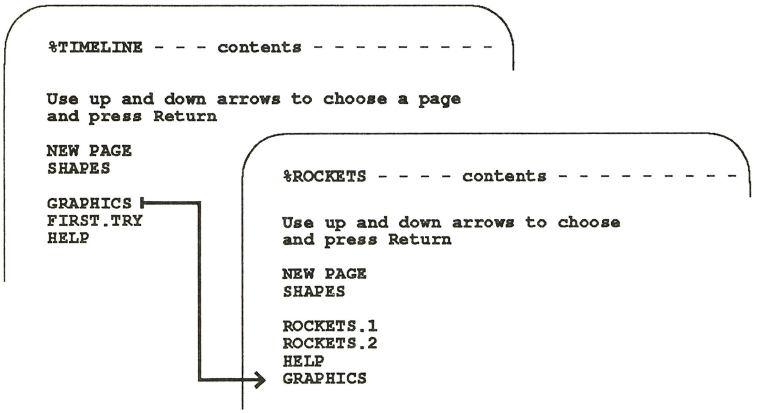
```
copyfile  "graphics  "graphics2
```

copies the file named **graphics**, and names the copy **graphics2**.



Since the file being copied is inside the current directory, and the new file will be inside the current directory, only the file names themselves are required. If you want to copy a file from another directory, or to copy a file from the current directory into a different directory, you would need the full pathname for both files.


The same volume name, since both directories are on the same disk.
copyfile "%scrapbk%timeline%graphics "%scrapbk%rockets%graphics
A different directory name.
The file name itself can be the same if it's in a different directory.


%TIMELINE - - - contents - - - - - - - - -
Use up and down arrows to choose a page and press Return
NEW PAGE
SHAPES
GRAPHICS
FIRST.TRY
HELP
%ROCKETS - - - - contents - - - - - - - - -
Use up and down arrows to choose and press Return
NEW PAGE
SHAPES
ROCKETS.1
ROCKETS.2
HELP
GRAPHICS

ProDOS Hierarchy and LogoWriter Commands

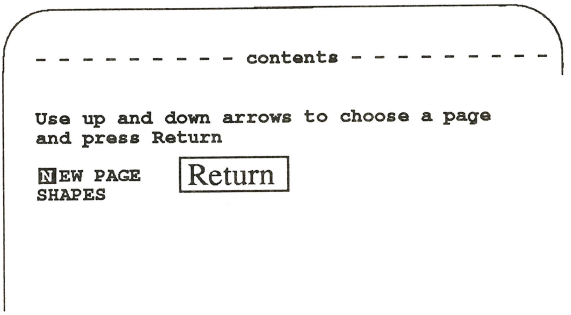
Pathname Hierarchy	Common Term	LogoWriter Commands and Reporters
Volume name	The name of the disk	To change volumes: **setdisk** or **setprefix** To find the name of the current disk: **show prefix** To find the names of the disks that are in the disk drives: **show online**
Directory name	The name of a group of files	To change directories: **setprefix** To find the name of the current directory: **show prefix** To find the names of the directories in the current directory: **show directories**
File name	The name of a file, such as: • a LogoWriter page • a text file • a graphics file • a program file	To get a LogoWriter page: **getpage** To load a graphics file: **loadpic** To load a text file: **loadtext** To save a graphics file: **savepic** To save a text file: **savetext** To find the names of LogoWriter pages in the current directory: **show pagelist** To find the names of the files in the current directory: **show filelist**

Example 1: Creating a Directory

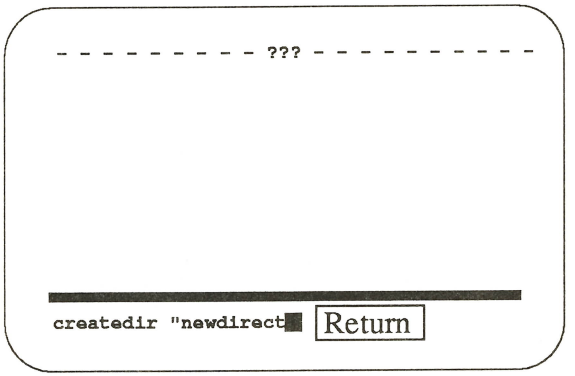
To create a directory, use the **createdir** command. **Createdir** needs an input, which represents exactly where the directory will be created.

If you use a simple input (just the name of the directory with no % signs), the directory will be created within the current directory.

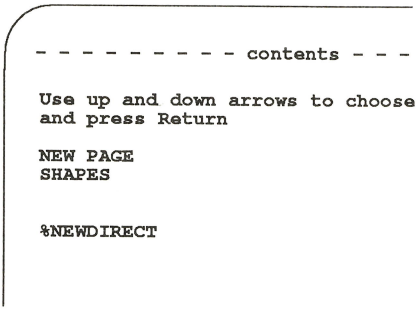
1. Choose New Page.



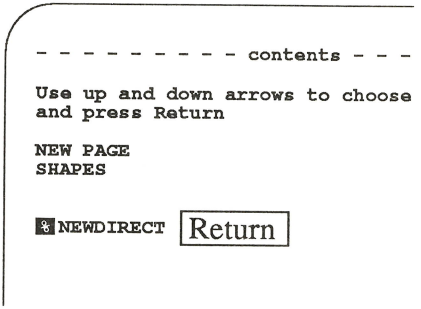
2. Create the new directory with the **createdir** command.



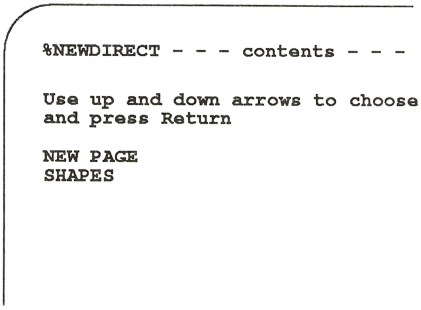
3. Press **Escape** to see the Contents page. The new directory is listed in Contents.



4. Open the directory by putting the cursor on it and pressing **Return**.



5. Use the subdirectory just like a scrapbook. To go back to the main directory, press **Escape**.



You can create a subdirectory within a subdirectory if you want. The only limit to the amount of subdirectories inside other subdirectories is that an entire pathname can't be longer than 64 characters.

Example 2: Erasing a Directory

Before you can erase a directory, it must be empty of pages and files. ProDOS cannot erase a directory that isn't empty.

1. Erase the pages. From the directory's Contents page, use [ó - 6] (the **erase to end of line** keys) to erase each page, one by one.

2. Erase the files. A directory can contain "hidden" files that can't be erased from the Contents page. For instance, there is probably a file for the Shapes page. There may also be text or graphics files from different programs.

   Go to a new page. Type:

   ```
   show  filelist
   ```

   If the only file name printed is **[SHAPES lws]**, then the only file you have to erase is Shapes, a LogoWriter system (lws) file.

   ```
   erasefile  "shapes
   ```

   Now press **Escape** to go to the directory's Contents, and press **Escape** again to leave the directory. The directory name will be in the Contents page. Put the cursor on it and press [ó - 6].

II LogoWriter 2.0 for IBM Computers

The features described in this chapter are those which are unique to the version of LogoWriter for IBM computers, and which can't be properly described in terms of primitive definitions. These are:

- LogoWriter Keys
- Game Port Inputs
- LogoWriter Graphics Colors
- The Vertical Bar Delimiter Character
- DOS Features: Subdirectory and File Management

LogoWriter Keys

Function Keys

F1	Select
F2	Cut
F3	Copy
F4	Paste
F6	Erase to End of Line
F8	Label
F9	Turtle-Move
F10	Help

Ctrl - Combination Keys

To use these keys, hold down the **Ctrl** key and press the key.

Ctrl-U	Up
Ctrl-D	Down
Ctrl-F	Flip
Ctrl-Break or	
Ctrl-S	Stop
Ctrl←	Beginning of Line
Ctrl→	End of Line

Special Keys

PgDn	Next Screen
PgUp	Previous Screen
Home	Top of Page
End	Bottom of Page
Insert	Open a New Line

Game Port Inputs

LogoWriter for IBM computers accepts two pairs of game paddles or two joysticks plugged into the game ports.

Note: The use of the game port primitives requires a game port adapter card.

Paddle reads the position of a game controller, and reports it as a number. The range of numbers that **paddle** reports depends on the type of joysticks or paddles being used. The input number to **paddle** directs it to the appropriate paddle or one axis of the joystick.

Game Paddles

The fully counterclockwise position reports the lowest number; fully clockwise reports the highest number.

Paddle 0 reads paddle 0, plugged into the *first* game socket. **Paddle 1** reads paddle 1 plugged into the first game socket.

Paddle 2 reads paddle 0, plugged into the *second* game socket. **Paddle 3** reads paddle 1, plugged into the second game socket.

Joysticks

One joystick can be used in place of two game paddles. **Paddle 0** and **paddle 1** read the x and y axes of a joystick plugged into the first game socket; **paddle 2** and **paddle 3** read the x and y axes of a joystick plugged into the second game socket.

LogoWriter Graphics Colors

There are three IBM color graphics systems which LogoWriter supports. The kind of graphics system is determined by the computer's hardware: in most cases an internal adapter card runs the graphics system. To find out which system is used by your computer, consult your system manuals or check the card itself. Setting up your disk to run the appropriate graphics card is described in the *Setting Up LogoWriter* supplement booklet.

The color numbers for the various systems are more or less compatible: on each system the first four color numbers refer to the same colors. On the EGA and PS/2 graphics systems, the first 16 color numbers refer to the same colors. Transferring pages from system to system may result in some changes to the screen appearance. You cannot transfer pages from LogoWriter 1.1 to 2.0. You must first convert the files with a file conversion program.

Note: LogoWriter 1.1 graphics are CGA only, and do not run on any other graphics card. The color numbers for version 1.1 are not the same as for version 2.0 : refer to the chapter *LogoWriter 1.1* for those color numbers.

IBM Color Graphics Adapter

The IBM Color Graphics Adapter (CGA) is one of the most commonly available systems. Most graphics adapters can run CGA style graphics (color mode 4 hex), even if they have additional capabilities. For example, to use LogoWriter 2.0 on the IBM PC*jr*, the CGA setting must be used, and only four colors are displayed.

CGA color numbers are 0 through 3, but you can use any numbers. Numbers larger than 3 "wrap": **setc 4** is interpreted as **setc 0, setc 5** as **setc 1**, and so on.

Number	Color
0	Black (turtle is transparent)
1	White
2	Cyan
3	Magenta

The CGA has 16 background colors, which are the same as the EGA colors, below.

Enhanced Graphics Adapter

The IBM Enhanced Graphics Adapter is a newer color graphics system than the CGA system. It is capable of producing 16 high resolution colors (color mode 0D hex). The color numbers for turtle graphics are the same as the color number for the background.

Number	Color
0	Black (turtle is transparent)
1	Bright White
2	Light Cyan (turquoise)
3	Light Magenta (purple)
4	Red
5	Dark Blue
6	Orange
7	Light Green
8	Dark Gray
9	Pale Gray
10	Bright Cyan (turquoise)
11	Bright Magenta (purple)
12	Light Red
13	Pale Blue
14	Yellow
15	Bright Green

PS/2 Graphics

The IBM PS/2 series of computers is capable of 256 high resolution colors (color mode 13 hex). The first 16 colors are the same as the EGA colors. Above color number 16, various color intensities and shadings are available.

Number	Color
0 - 15	as EGA graphics

Each intensity group of colors (16 to 88, 89 to 160, 161 to 232) is in three saturation levels. Each saturation level ranges from blue to red to green.

16 - 88	**High intensity colors** 16 - 40 high saturation 41 - 64 medium saturation 65 - 88 low saturation
89 - 160	**Medium intensity colors** 89 -112 high saturation 113 - 136 medium saturation 137 - 160 low saturation
161 - 232	**Low intensity colors** 161 - 184 high saturation 185 - 208 medium saturation 209 - 232 low saturation
233 - 247	**Shades of gray**
248 - 255	**Black**

The Vertical Bar (|) Delimiter Character

The vertical bar character (on some keyboards the character looks like this: |; on others the bar is broken, like this: ¦) is used by LogoWriter as a special character to start and end a *character string*. Normally in LogoWriter a character string (also known as a *word*) starts and ends with a space. This separating of words by spaces is automatic and natural. It can in some cases become a problem, for instance in programs which want to print on the screen in special formats. For these needs, a way to turn off LogoWriter's automatic "delimiting" of words is required, hence the vertical bar. Any characters inside a pair of vertical bars are considered to be a character string - a single word - and the word can contain special characters which LogoWriter would normally use as separators (delimiters)*.

Consider the following examples:

```
print  "hello  there
I  don't  know  how  to  there
```

The first character string (the word **hello**) was used as input to **print**. The space between **hello** and **there** ended the string, and LogoWriter treated **there** as a command - one which is not defined, therefore the error message.

```
print  "|hello  there|
hello  there
```

The vertical bars extended the character string, even though a space separates the two words. The string inside the vertical bars can be indefinitely long - until a carriage return. A carriage return character automatically ends the string.

---

* In other versions of LogoWriter and Logo, the backslash character (\) functions to suppress delimiting, but it only works on one character at a time. The vertical bars define an entire string, which can be of almost any length.

DOS Features: Subdirectory and File Management

DOS stands for Disk Operating System*. It is the system through which the computer communicates with its disk drives for saving, loading,and organizing files on a disk. LogoWriter 2.0 contains primitives and functions which utilize some of the special features of the IBM computer's DOS. These features are somewhat advanced, but if you are using LogoWriter on a network or using a hard disk, you will probably want to take advantage of them.

This chapter is a general introduction, not a reference manual for either PC-DOS or MS-DOS®. For complete information on your version of DOS, refer to the manual that came with your computer.

Note: To set up your LogoWriter disk, you will have to use some DOS file copying commands. The complete instructions on the commands used to set up LogoWriter disks for various computer configurations are contained in the supplement on *Setting Up A LogoWriter Disk*, which is part of the LogoWriter package.

Files and Pages

A LogoWriter scrapbook contains pages. A "page" is a special kind of file which mixes text and graphics. LogoWriter's "page" commands work on LogoWriter pages and not on other kinds of files. The Contents page only lists pages; **pagelist** only lists pages; **getpage** only gets pages; **erpage** only erases pages.

A disk can contain many kinds of files other than pages. The DOS primitives are specifically intended to provide LogoWriter users with access to these other types of files; for example,to add the text from a different word processor into a LogoWriter page, or to use LogoWriter to create text files.

---

* IBM computers use PC-DOS; so-called "IBM compatibles" use MS-DOS. For our purposes they are the same.

In general, if you are working with LogoWriter pages, use the "page" commands: **getpage**, **erpage**, **newpage**, etc. If you want access to other kinds of files or IBM DOS filing features, use the "file" commands: **copyfile**, **erasefile**, **loadtext**, **savetext**, **loadpic**, **savepic**, and **rename**. All of these primitives are described in the chapter *IBM DOS Primitives*, in Section 2.

Note: For those who also use LogoWriter 2.0 for Apple computers, the ProDOS and IBM DOS commands have different names. This is to prevent confusion: the two systems are substantially different in their internal operations. However, the keystrokes used to enter and exit subdirectories from the Contents page are exactly the same on both kinds of computers, and the Contents page appearance is very similar.

Disks and Directories

LogoWriter generally refers to a disk by its location - that is, LogoWriter will get pages or save pages to the disk in the drive that LogoWriter started up on, even if you switch disks. The easiest way to copy a page from one disk to another is to switch disks while a page is showing on the screen and press Escape. The page is saved onto the new disk.

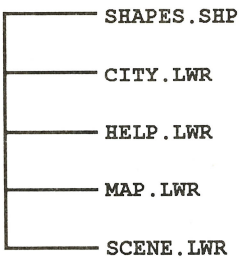
Using subdirectories means identifying a page's location more precisely. Using subdirectories adds tremendous flexibility to LogoWriter: you can organize your pages, and you can have more than one Shapes page on a disk. However, disk operations also become more complex.

Directories and Subdirectories

A directory is a file containing information on other files on the disk: their names, sizes and location. Every disk has at least one directory. The disk's main directory is called the "root" directory. It does not have a name, but is referred to by a special character, the backslash (\).

A LogoWriter scrapbook disk without subdirectories

The root directory contains the files



The Contents page would look like this:

```
------------contents------------

Use ↑ and ↓ keys to choose a page
and press Enter

NEW PAGE
SHAPES
CITY
HELP
MAP
SCENE
```

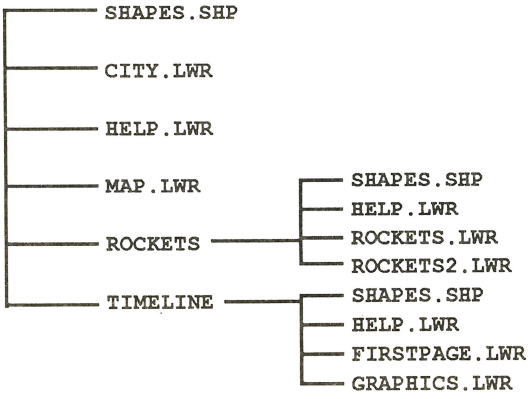
A disk can optionally contain subdirectories. A subdirectory is a directory which is contained in another directory. Each subdirectory acts like a disk on its own: in LogoWriter, each subdirectory has its own Contents page and its own Shapes page.

When LogoWriter uses subdirectories, each subdirectory becomes, in effect, a complete LogoWriter scrapbook.

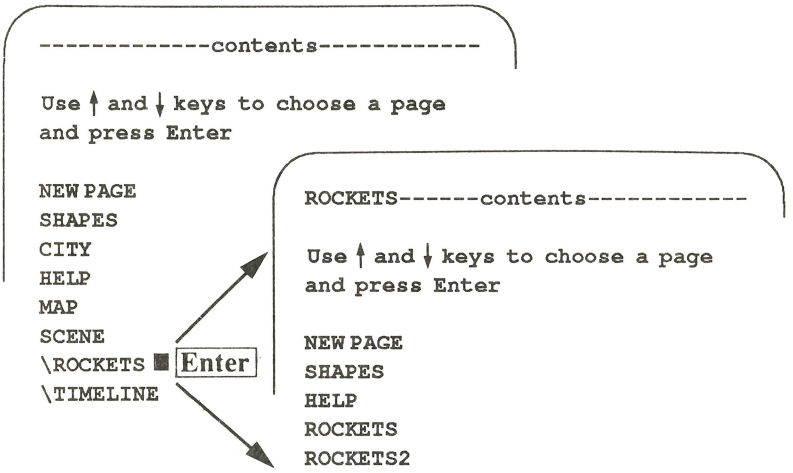
A LogoWriter disk, using subdirectories

The root directory contains pages and subdirectories

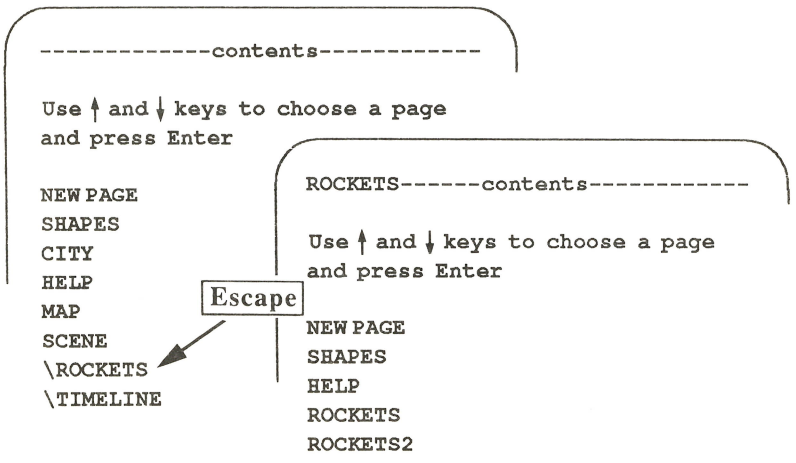
Pages inside subdirectories



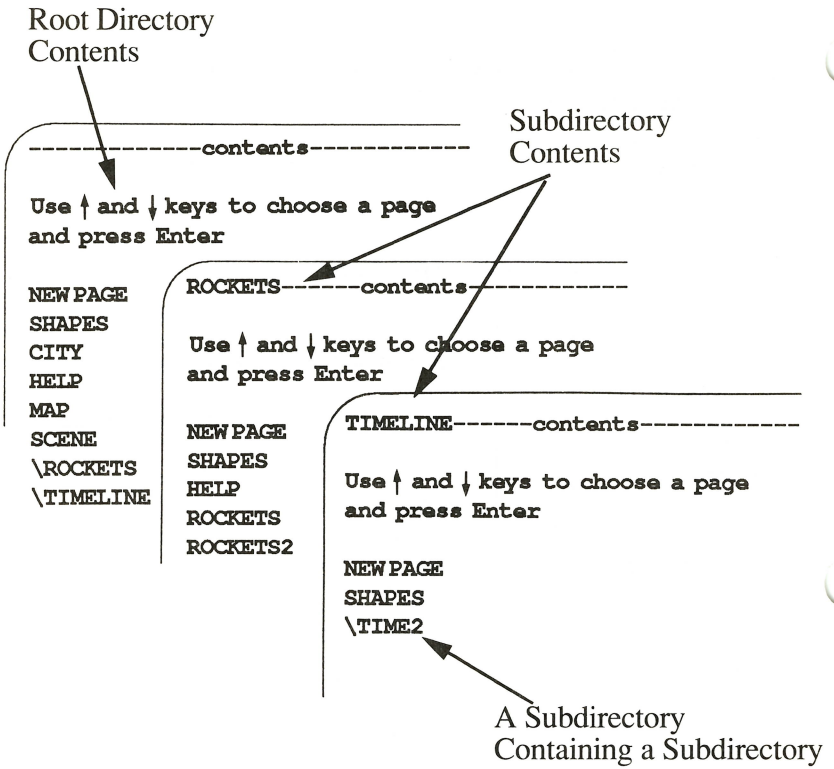
You can use a subdirectory to store a completed project on a disk, with its own Shapes page. A teacher can keep a disk of class projects, with a subdirectory for each student. Subdirectories show on Contents like pages, but their names start with a backslash character. To open a subdirectory, put the cursor on the subdirectory name and press **Enter**.



To leave a subdirectory, press **Escape** from the subdirectory's Contents page.



A disk can contain many subdirectories. Subdirectories can contain subdirectories, to quite a number of levels.



Summary

Open a LogoWriter subdirectory from the Contents page by putting the cursor on the subdirectory name and pressing **Enter**. Leave a subdirectory from the subdirectory's Contents page by pressing **Escape**.

There are also LogoWriter primitives to change directories, and to access files in other subdirectories. See the chapter *IBM DOS Primitives*.

A Special Note About IBM DOS

If you are using subdirectories, and use **setdisk** to change the disk drive for saving and loading, the computer will remember the directory you were using when you changed disk drives. When you use **setdisk** to return to the disk, you will automatically return to the subdirectory you left. In this way, IBM DOS is different than ProDOS, the Apple *II* computer's operating system. In ProDOS, **setdisk** returns to the root directory.

Creating and Erasing Subdirectories

LogoWriter has been designed so that you can enter and exit subdirectories by pressing keys on the Contents page, without any need for typing commands with DOS syntax. To create and erase directories does require commands. These commands can be used with simple inputs or with the more complex "pathname" syntax. For the sake of clarity and ease of use, these examples use simple inputs without special DOS characters.

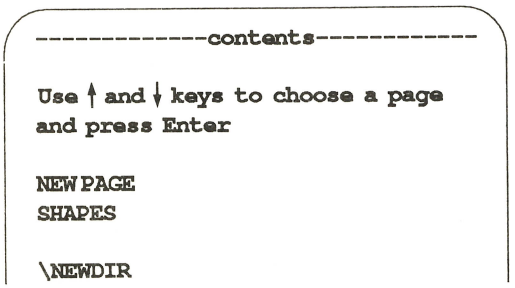
Creating A Directory

To create a directory, type **mkdir**:

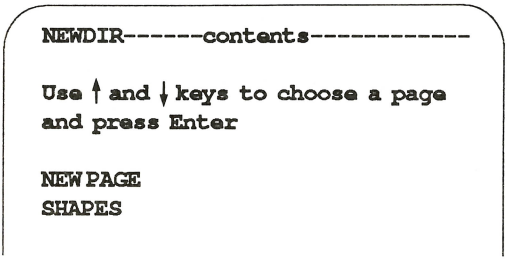
```
mkdir  "newdir
```

Mkdir creates a new directory within the current directory.

To see the subdirectory, press Escape to leave the page. **Newdir** is listed in the Contents page.



To enter **newdir**, put the cursor on the name and press Enter. You will see the **newdir** Contents page.



Note: There is a listing for a Shapes page, even though the directory is empty. LogoWriter holds the turtle shapes in memory at all times, and it is always possible to see and edit turtle shapes. When you get the Shapes page and press Escape to leave it, those shapes are saved in a Shapes file in the directory. Any changes to those shapes will not affect the shapes in other directories.

To create a subdirectory within **newdir**, go to a new page in the subdirectory and use **mkdir**. **Mkdir** always creates a directory within the current directory, unless you use special DOS characters to specify the path in which the directory will be created. Refer to your DOS reference manual for more information on making pathnames with the backslash character.

Erasing a Subdirectory

If a subdirectory is empty, it can be erased from the Contents page in the same way as a page: by putting the cursor on its name and pressing F6. A directory that contains files can't be erased.

To empty a directory, erase every file that it contains. If you want to keep some pages, copy them into another directory with **copyfile** before erasing them.

1. Go to the directory's Contents page and erase each page by pressing F6 on the page name.
2. Choose New Page. In the new page's command center, type:

```
show  filelist
```

If there are any files left in the directory, **filelist** will list them. Often there will be a Shapes page left. Erase the file(s) with **erasefile**. You must use a complete filename as input to **erasefile** - including the filename extension characters. To erase a Shapes page, type:

```
erasefile  "shapes.shp
```

Leave the page without saving it, by typing:

```
leavepage
```

On the directory's empty Contents page, press Escape to go to the previous directory's Contents. Put the cursor on the name of the empty directory and press F6.

Using Commands to Change Directories and Copy Pages

There are two ways to copy a page from one directory to another. One way is to get the page on the screen and then use the **chdir** command to change directories. After changing directories, press Escape to save the page in the new directory.

```
getpage  "pageone
show  directories
[DIR1  DIR2]
chdir  "dir1
```

...and press Escape to save the page.

Note: To copy onto a subdirectory on another disk, use **setdisk** followed by **chdir**, then press Escape.

You can also copy any kind of file, including pages, with the **copyfile** command. To copy a file from one directory to another requires the use of pathnames: the name of the directory as well as the name of the file must be combined into one input, using the backslash character to connect them. The following command would copy a LogoWriter page named **pageone** from the subdirectory named **dir1** into the subdirectory named **dir2**.

```
copyfile  "\dir1\pageone.lwr
"\dir2\pageone.lwr
```

Special DOS Characters for Chdir

There are a couple of special characters that can be used with **chdir**. They provide quick and useful shortcuts.

The backslash character (\), used alone as input, goes to the disk's root directory, no matter what directory is current.

```
show  currentdir
work
chdir  "\
show  currentdir
\
```

Two periods (..) tell DOS to "back up" one level of subdirectory. In the following example, the subdirectory **work** is contained in the subdirectory **john**.

```
show  currentdir
work
chdir  "..
show  currentdir
john
```

Subdirectories and LogoWriter Commands

The following table lists all the DOS commands and reporters, and relates them to commonly used DOS terminology.

DOS Hierarchy	Common Term	LogoWriter Commands and Reporters
Root directory	The main list of disk files	To go to the root directory: `chdir "\` To change disk drives: `setdisk` To find the current disk drive: `show disk`
Subdirectory	A group of files	To change directories: `chdir` To find the name of the current directory: `show currentdir` To find the subdirectories in the current directory: `show directories`
File name	The name of a file, such as: • a LogoWriter page • a graphics file • a text file	To get a LogoWriter page: `getpage` To load a graphic file: `loadpic` To load a text file: `loadtext` To save a graphic file: `savepic` To save a text file: `savetext` To find the LogoWriter pages in the current directory: `show pagelist` To find the files in the current directory: `show filelist`

III LogoWriter 1.1

About LogoWriter 1.1

LogoWriter 1.1 replaced the original LogoWriter 1.0. It is in every way compatible with LogoWriter 1.0 and contains some new features and a number of corrections over LogoWriter 1.0. There is a version of LogoWriter 1.1 for the Apple *II* family including the *II*+, for the IBM PC, the IBM PC*jr*, and for the COMMODORE 64.

LogoWriter 2.0 has, in most cases, replaced LogoWriter 1.1. LogoWriter 2.0 is a completely new release of LogoWriter and is not file compatible with previous versions. File conversions, however, are easy. LogoWriter 2.0 is available in versions for the Apple *II* family of computers and IBM PC's, including the PS/2 computers.

Why Use LogoWriter 1.1?

For Apple *II* computers, LogoWriter 1.1 requires 64K of memory, whereas LogoWriter 2.0 requires 128K. So, owners of 64K Apple *II*e computers must either upgrade their computer's memory or use LogoWriter 1.1. Owners of Apple *II*+ computers can use LogoWriter 1.1, and must directly request the modified LogoWriter 1.1 for the Apple *II*+.

For IBM PC computers, LogoWriter 1.1 requires 128K of memory, whereas LogoWriter 2.0 requires a minimum of 256K (and will take advantage of more if it is available). So LogoWriter 1.1 can be the proper choice for those computers with less than 256K of memory.

The IBM PC*jr* is specially supported by LogoWriter 1.1 and many PC*jr* users will continue using it. LogoWriter 2.0 can run on the PC*jr*, but only displays 4 colors using the CGA color mode, and requires a 128K memory expansion to increase the memory to 256K, as well as a special disk setup to utilize the extra memory.

For the COMMODORE 64, the only version of LogoWriter is version 1.1.

Last Page

LogoWriter 1.1 has a special page called Last Page which contains a backup copy of the last page that was saved.

This is a precaution against the accidental loss of a page; for instance, a page can be lost by typing **cp** and pressing Escape, or by pressing Escape accidentally.

Immediately after saving a page accidentally, you can recover the previous version of the page by choosing Last Page from Contents. Immediately name the page with **namepage** and press Escape to save it.

LogoWriter 2.0 does not have a Last Page feature. Instead, **cp** has been changed so that it goes to a new, clean page, and page locking (**lock** and **unlock**) was added to protect pages from being changed or erased.

LogoWriter 1.1 for Apple II Computers

LogoWriter 1.1 and 2.0 use the same keys, the same color numbers, and access game paddles in the same manner in the Apple *II*e™, *II*c™ and IIGS™. For the keys and colors, see Section 3: LogoWriter Versions, where version 2.0 is described.

Memory Requirements

LogoWriter 1.1 for Apple *II* computers runs on any Apple *II*e, *II*c, or IIGS computer with 64K of memory. It does not run on the smaller 3 1/2" disks on the *II*GS; the larger 5 1/4" disks are required.

LogoWriter DOS

LogoWriter 1.1 uses a unique disk operating system called LogoWriter DOS. Disk files are in a unique format that is not accessible to DOS 3.3 or to ProDOS. The only way to copy files, copy disks, or to format blank disks is to use the LogoWriter version 1.1 Disk Manager program.

Converting Pages

It is possible to convert pages from version 1.1 to version 2.0 (which uses Apple ProDOS disk operating system). The LogoWriter version 2.0 Disk Manager contains options to convert pages to and from LogoWriter DOS and ProDOS.

LogoWriter for the Apple II+

A special version of LogoWriter 1.1 which runs on the Apple *II+* is available on special order. It requires a 64K Apple *II+* with at least one disk drive.

LogoWriter on the *II+* runs exactly the same as on a *IIe*, with the obvious differences in keyboards and upper and lowercase characters.

Making Copies

To make new scrapbook disks, or to change the printer setting, use the version 1.1 Disk Manager on the LogoWriter 1.1 Master Disk. If you have a site license, you can also use the version 1.1 Disk Manager to make copies of LogoWriter disks. You need an Apple *IIe*, *IIc* or IIGS to run the Disk Manager.

To copy the Apple *II+* LogoWriter disk, use the "Copy a Disk" option, *not* "Create a LogoWriter Disk" (which will create a version for the *IIe* keys).

Exchanging Disks

Apple *II+* scrapbook disks can be freely interchanged with Apple *IIe* and *IIc* version 1.1 scrapbooks. Although the Apple *II+* cannot display upper and lowercase letters in the command center or on the flip side of the page, LogoWriter does show upper and lowercase on the page itself, and it "remembers" the upper and lowercase state of letters that are cut from the page to the flip side, or transferred from another Apple to a *II+*.

Key Use

The key use is changed. There are no **up** and **down arrow** keys or Apple keys on the Apple *II+*. You can, however, display upper and lowercase characters on the page. With the common "shift key modification" (a jumper wire connecting the **Shift** key to pin 4 of the joystick plug), you can use the **Shift** key to shift upper and lowercase characters, or you can use the **Shift** and **Caps Lock** key combinations.

The Apple key is replaced by CTRL-A. Press CTRL-A, then let go and press the number or letter.

CTRL-A	U	Up
CTRL-A	D	Down
CTRL-A	F	Flip
CTRL-A	1	Select
CTRL-A	2	Cut
CTRL-A	3	Copy
CTRL-A	4	Paste
CTRL-A	6	Erase to End of Line
CTRL-A	8	Label
CTRL-A	9	Turtle-Move
CTRL-A	0	Help
CTRL-A	→	Next Screen
CTRL-A	←	Previous Screen
CTRL-G		Stop keys

The **arrow** keys and others are given **Control** -key combinations:

CTRL-K	Up arrow
CTRL-J	Down Arrow
CTRL-B	Backspace/Delete
CTRL-D	Delete under cursor
CTRL-S	Shift the next key
CTRL-C	Caps Lock

Other special LogoWriter characters:

CTRL-Q	Backslash \
Shift-N	Left bracket [
Shift-M	Right bracket]

LogoWriter 1.1 for the IBM PC and IBM PC*jr*

There are two versions of LogoWriter for IBM computers: one for the PC*jr* and one for the IBM PC. The main differences are the number of screen colors, and the special keys on the keyboard.

Common Features

The IBM PC and PC*jr* share many features, including a special buffer for making sounds, number of disk drives, and access to game paddles.

Making Sounds

Version 1.1 for the IBM PC and PC*jr* has a feature called a "tone buffer." The sound channel has its own area of memory from which the sound-making commands are played, and the rest of LogoWriter doesn't have to wait for sounds to finish before going on to do something else.

```
repeat 10 [tone 1000 30 fd 30]
```

The tones will keep playing long after the turtle has stopped. If this is a "bug" in the context of a program, adding a **wait** command will make LogoWriter wait the same length of time as the tone takes:

```
repeat 10 [tone 1000 30 fd 30 wait 30]
```

If you want a program to wait until a long sound or melody has finished playing, use the **tone?** primitive. **Tone?** reports **true** if there is still a tone left in the tone buffer, and **false** if there isn't. The **waitforquiet** procedure will keep waiting until there is no tone playing:

```
to waitforquiet
if not tone? [stop]
wait 1
waitforquiet
end
```

Try **waitforquiet:**

```
repeat 4 [tone 400 30] print [not
finished]
repeat 4 [tone 400 30] waitforquiet pr
[finished]
```

IBM PC LogoWriter Colors

LogoWriter 1.1 can display color graphics on a color monitor, but it only supports CGA-style graphics. If your computer uses a different graphics system, such as EGA graphics or PS/2 graphics, you must use LogoWriter 2.0. The color numbers are 0 through 3, but you can use any numbers. Numbers larger than 3 "wrap": **setc 4** is interpreted as **setc 0, setc 5** as **setc 1**, and so on.

Important: The color numbers for LogoWriter 2.0 are not the same as those below. This can be important if you will be converting programs from 1.1 to 2.0 or vice versa.

Number	Color
0	Black
1	Cyan
2	Magenta
3	White

The IBM PC has 16 background colors:

Number	Color
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	White
8	Gray
9	Light Blue
10	Light Green
11	Light Cyan
12	Light Red
13	Light Magenta
14	Yellow
15	Bright White

LogoWriter 1.1 for the IBM PC*jr*

LogoWriter for the PC*jr* runs on all standard IBM PC*jr* computers, with a minimum of 128K memory.

Fn - Combination Keys

To use any of these keys, press the **Fn** key, then press the key.

Fn - 1	Select
Fn - 2	Cut
Fn - 3	Copy
Fn - 4	Paste
Fn - 6	Erase to End of Line
Fn - 8	Label
Fn - 9	Turtle-Move
Fn - 0	Help
Fn - B	Stop
Fn - →	Next Screen
Fn - ←	Previous Screen
Fn - ↑	Top of Page
Fn - ↓	Bottom of Page

Ctrl - Combination Keys

To use any of these keys, hold down the Ctrl key and press the key.

Ctrl - U	Up
Ctrl - D	Down
Ctrl - F	Flip
Ctrl - ←	Beginning of Line
Ctrl - →	End of Line

PC*jr* LogoWriter Colors

Number	Color
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	White
8	Gray
9	Light Blue
10	Light Green
11	Light Cyan
12	Light Red
13	Light Magenta
14	Yellow
15	Bright White

LogoWriter 1.1 for the COMMODORE 64 Computer

Note: The COMMODORE 64 computer has no key labeled **Escape**. Instead, the **F1** key functions as the **Escape** key.

COMMODORE - Combination Keys

To use any of these key combinations, hold down the C= key and press the key.

C= - 1	Select
C= - 2	Cut
C= - 3	Copy
C= - 4	Paste
C= - 6	Erase to End of Line
C= - 8	Label
C= - 9	Turtle-Move
C= - 0	Help
C= - U	Up
C= - D	Down
C= - F	Flip
C= - S	Stop
C= - CRSR→	Next Screen
C= - ←	Previous Screen
C= - ↑	Top of Page
C= - CRSR	Bottom of Page

C= - B	Beginning of Line
C= - E	End of Line
C= - T	Tab
C= - O	Open a Line

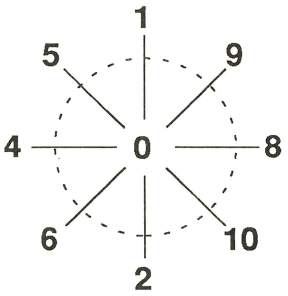
COMMODORE 64 LogoWriter Colors

The COMMODORE 64's color numbers are 0 through 15, but you can use any numbers. Numbers larger than 15 "wrap": **setc 16** is interpreted as **setc 0**, **setc 17** as **setc 1**, and so on.

Number	Color
0	Black
1	White
2	Red
3	Cyan
4	Purple
5	Green
6	Blue
7	Yellow
8	Orange
9	Brown
10	Light Red
11	Dark Gray
12	Gray
13	Light Green
14	Light Blue
15	Light Gray

Game Paddle Inputs

LogoWriter for the COMMODORE 64 computer accepts one joystick. The joystick must be plugged into the game port socket 2. The **paddle** primitive reads the position of the joystick, and reports it as one of the numbers indicated in the following diagram.



The **paddle** primitive will only accept the numbers 1 or 2 as input. These numbers specify which port is to be read.

Warning: If the input to **paddle** is 1, random characters will appear on the screen. Plug the joystick into port 2 only.

Section 4: Appendices

This section contains five appendices covering technical issues and special LogoWriter features.

Appendix 1, *LogoWriter Primitives and Special Words*, lists the names of all of the LogoWriter primitives. They are listed in groups according to function: graphics primitives, word processing primitives, and so on. As your LogoWriter expertise grows, you may want to consult this list to find special purpose primitives, or primitives which are related to those you know.

Appendix 2, *Startup*, describes the two uses of the special name "startup," with suggested uses.

Appendix 3, *Tools*, covers the advantages of LogoWriter's tools feature, with tips on using tools in your programs.

Appendix 4, *LogoWriter Messages*, lists and describes every message that LogoWriter prints when it encounters an error or computer hardware problem.

Appendix 5, *Glossary of LogoWriter and Computer Terms*, is a list of terms used in the LogoWriter documentation or in other books on the Logo language.

Appendix 1
LogoWriter Primitives and Special Words

@ Indicates a primitive which is exclusive to LogoWriter for IBM computers.

* Indicates a primitive which is exclusive to LogoWriter for Apple *II* computers.

Screen

cc
cg
cp
ct
nextscreen
prescreen
* undo

Scrapbook

cleartools
contents
erpage
flip
front?
getpage (gp)
getshapes
gettools
leavepage
load
lock
namepage (np)
newpage
pagelist
restore
savepage
shapes
toollist
unlock

Workspace

recycle
space

Assigning

clearname
clearnames
make
name
name?
printnames
shownames
thing

Flow of Control/Logic

and
if
ifelse
not
or
output (op)
repeat
run
stop
stopall

Events

clearevents
when

Graphics

all
ask
back (bk)
bg
charunder
clean
color
colorunder
cursorpos
@ distance
each
@ fastturtle
fill
forward (fd)
heading
home
ht
label
left (lt)
pd
pe
pos
pu
px
rg
right (rt)
setbg
setc
seth
setpos
setsh
setx
sety
shade
shape
@ slowturtle
st
stamp
tell
towards
who
xcor
ycor

Printer Commands

dspace
printscreen
printtext
printtext80
sspace

Text Editing/Words and Lists

ascii
bottom
butfirst (bf)
butlast (bl)
cb
cd
cf
char
clipboard
copy
count
cu
@ cursorchar
cut
delete
empty?
eol
equal?
first
found?
fput
identical?
insert
item
last
list
list?
lput
member?
number?
parse
paste
print (pr)

replace
search
select
selected
sentence (se)
@ settc
settextpos
show
sol
tab
@ tc
textlen
textpos
top
type
unselect
word
word?

Pausing

wait

Sound

tone

Math

+
-
/
*
=
>
<
arctan
cos
int
minus
random
remainder
round
sin
sqrt

Input

button?
key?
paddle
readchar
readlist (rl)
readlistcc (rlcc)

Special Words

end
false
startup
to
true

Disk

disk
setdisk

ProDOS

* bye
copyfile
* createdir
directories
erasefile
filelist
loadpic
loadtext
* online
* prefix
rename
savepic
savetext
* setprefix
* setslot
* slot

IBM DOS

@ chdir
copyfile
@ currentdir
directories
@ dos
erasefile
filelist
loadpic
loadtext
@ mkdir
rename
@ rmdir
savepic
savetext

System

* .bload
* .bsave
* .call
.deposit
.examine
@ .in
@ .out
@ .primitives
.version

Appendix 2
Startup

In LogoWriter *startup* is a special word. It can be used as a page name and as a procedure name.

Startup as a Page Name

Any scrapbook can have a page named Startup. When LogoWriter starts up, it will automatically get the Startup page, without going to Contents.

You can make LogoWriter run the Startup page on any disk, not just the disk that LogoWriter starts up on. When the splash screen appears, change the disk in drive A (drive 1). LogoWriter will run the Startup page on the disk in drive A.

Startup can be used to:

- Display an individualized welcome page.
- Display a page of text; for example, instructions for a class activity.
- Load tool procedures or variables into the workspace to "pre-set" a specific learning environment.

The *Startup Your Scrapbook* activity card (purple section of Intermediate activity cards), describes one application of the Startup page name.

Startup as a Procedure Name

Any page can contain a procedure named **startup.** If there is a procedure named **startup**, it will run as soon as the page is displayed.

The **startup** procedure runs after a **getpage** command, or after the page is chosen from Contents.

The **startup** procedure does *not* run after a **load** or **gettools** command.

The *Bring Your Page to Life* activity card (red section of Intermediate activity cards), describes an application of the **startup** procedure name.

Startup Hints

- There can be only one **startup** procedure on a page. If by mistake there are two, LogoWriter will only run the last one on the page.

- Use a **startup** procedure to load variables (the **make** or **name** primitives can be used to create variables):

```
to  startup
name    0   "turtle
name   11   "square
name   12   "circle
name   21   "cat
name   22   "rabbit
name   23   "tree
name   24   "flower
name   25   "helicopter
contents
end
```

Define the procedure on the page named Startup. When LogoWriter starts up, it will display the Startup page and automatically run the **startup** procedure. When the variables have been assigned their values, LogoWriter will display the Contents page.

Appendix 3
Tools

What are Tools?

Tool procedures are LogoWriter procedures which reside in the computer's memory only. They do not show on the flip side of a page and they are not cleared from the workspace when pages are changed. In this way, they act like primitives, although they are really LogoWriter procedures. Any page of procedures can be used as tools by putting them into memory with the command **gettools**. Putting procedures into memory as tools is very useful when you do not want the procedures to be erased easily. Another advantage is that memory space is used more effectively; since tools do not show on the flip side, they do not consume text space. Many more procedures can be in memory at one time than can fit on the page at one time. This is especially true of LogoWriter 2.0, which has a greatly expanded workspace.

Advantages of Tools

Using tools has several advantages:

- They remain in the workspace when pages are changed.
- They are not easily erased by accidental editing.
- Tools don't take up space on the flip side of the page, leaving more space for other procedures.

Any page of procedures can be used as tools by loading the page with **gettools**.

Making a Page of Tools

Any page of procedures can become a page of tools. **Gettools** loads all the procedures on a page as tools. **Cleartools** removes them from the workspace.

Define some procedures on a new page:

```
to sqr :x
output :x * :x
end

to cube :x
output :x * :x * :x
end
```

Test that you have typed the procedures correctly. Name the page **tooltest**, and press Escape. On the Contents page, choose New Page, then type:

```
gettools "tooltest
show sqr 5
25
show cube 3
27
```

Sqr and **cube** are now tool procedures. Go to the flip side of the page. **Cube** and **sqr** are not there. Use **toollist** to find out the names of the tool procedures in the workspace.

```
show toollist
[sqr cube]
```

Debugging Tools

Tool procedures should be fully debugged before they are used, since it's impossible to edit them when they are hidden. To change or debug tool procedures, get the tools page with **getpage** or from Contents, so the tools become "regular" LogoWriter procedures on the flip side of the page.

Duplicate Procedure Names

Avoid using the same name for a tool procedure and a procedure on the current page. If a tool procedure and a regular procedure have the same name, the regular procedure will replace the tool procedure. For example, if a procedure named **circle** is on the flip side of a page, and another **circle** is loaded as a tool, the tool **circle** is cleared from memory. If **circle** is typed in the command center, LogoWriter will run the **circle** on the flip side of the page.

In the same way, if you have put the page Mathtools in memory with **gettools**, and then choose Mathtools from Contents, the procedures will cease being tools. By appearing on the flip side, they will become "normal" procedures.

Get Tools With Startup

Use the startup features to automatically load tools. Create a page named Startup, with this procedure on it:

```
to startup
gettools "arcs          Use the name of a tools
contents                page.
end
```

Press Escape to save the page. Restart LogoWriter. When LogoWriter starts up, the Startup page is displayed first. The **startup** procedure runs, loading the tools. When the tools have been placed in the workspace, the Contents page is displayed.

Tools Pages on the Master Disk

Tools pages are provided on the LogoWriter Master Disk. With some versions of LogoWriter, they can be copied with the Disk Manager program. With other versions, you can get them as regular pages and save them onto any disk in the usual fashion. There are different tools pages for each version of LogoWriter, and these pages change as we update them to provide more useful tools.

We recommend that you examine these tools by getting the pages as regular pages, i.e., from Contents or with **getpage**. Choose the procedures that you find most useful, and copy those procedures onto different pages. If you want to copy only one tool onto another page, be sure to check whether it needs subprocedures, and if so, copy them as well. On the flip side of each page, comments are written in between procedures, describing the procedures, their function, and whether they need subprocedures.

Appendix 4
LogoWriter Messages

This is a complete listing of the messages that LogoWriter can print. Some of the messages do not appear in all versions of LogoWriter.

) in wrong place
A parenthesis was used incorrectly - check that all parentheses are in matched pairs, with the same number of opening parentheses as closing ones.

Can't divide by zero
A number cannot be divided by zero. Check the divisor you used. It may be a small enough decimal to truncate to zero, or a variable that has the value of zero.

Can't fill or shade
A shape with many sides is too complex for LogoWriter to fill or shade.

Can't use *character* in a page name
On some computers, the disk filing system doesn't allow certain characters to be used in page names.

Can't use *name* as a page name
Some names you choose may be names which the computer's operating system reserves for special purposes.

Can't use to as a command
The special word **to** was used incorrectly: either it was typed in the command center, or it was put inside a procedure. Check to see if you forgot to put **end** at the end of the previous procedure. **To** is only used to start a procedure definition.

Directory is not empty
A directory cannot be erased if it contains files. Use **filelist** to find the files in the directory, and then erase each file.

Directory not found
A **setprefix** command was directed to a directory that doesn't exist (probably because of a typing error in the ProDOS pathname), or the disk was switched following a **setprefix** command. If you choose to use directories and the prefix method of using ProDOS, you must use **setprefix** to save onto a different disk, instead of just switching the disks in the drive. Use **online** to find the name of the disks in the disk drives, and **prefix** to find the prefix setting, then use **setprefix** to set LogoWriter to refer to one of the current disks. Alternately, you can use the simpler system of saving to any disk that is in a disk drive: use **setdisk "a** or **setdisk "b**, and you can switch disks without generating this error.

Disk full or Disk or directory full
There is no more room on the disk or in the directory you are using. If the page you are on has not been saved, switch to another disk to save the current page. In ProDOS, a directory can contain 51 pages, but a disk can contain more: switch to another directory.

Disk is write-protected
You can't save any pages on a disk that is write-protected. Switch disks, or remove the disk and remove the write-protect sticker (with 3.5" disks, slide the write-protect tab).

DOS prevents that operation
You have tried to use an incorrect file extension in a file or directory name, and DOS prevents access to the file name or pathname. Use a different file name or pathname.

I don't know how to *name*
LogoWriter tried to run a word that is neither the name of a primitive, a tool, nor a procedure on the page.

I don't know what to do with *name*

A line (typed in the command center or used in a procedure) began with a reporter. Every instruction must begin with a command.

I'm having trouble with the disk or drive

Either the disk is write-protected, there is no disk in the drive, the drive door is open, or there is some other problem of this type. Check the disk and the drive before continuing.

I'm having trouble with the printer

The printer may not have the "select" button on, or may have another problem, such as a paper jam or no paper. Check that a printer is connected, then check the paper and switches on the printer. If there is no visible problem, you may have to change the printer setting on your LogoWriter disk.

Joystick or paddle not connected

Check that the paddles or joystick are plugged in correctly. Some computers require a game port adapter card for the use of joysticks and paddles.

Missing] in *name*

Check that all square brackets in the procedure *name* are in matched pairs, with the same number of opening brackets as closing ones. This message usually occurs when LogoWriter is putting procedures into the workspace (e.g., when it is flipping from the flip side, getting a page, or loading a page).

Missing [in *name*

Check that all square brackets in the procedure *name* are in matched pairs, with the same number of opening brackets as closing ones.This message usually occurs when LogoWriter is putting procedures into the workspace (e.g., when it is flipping from the flip side, getting a page or loading a page).

Name can only be used in a procedure
Stop or **output** was used in the command center. These commands are used to stop procedures, so they can only be used inside procedures.

Name didn't report anything to *name*
A primitive or procedure requiring an input didn't get one from the primitive or procedure following it.

Name doesn't like *name* as input
Some primitives require a certain type of input: for example, math operations require numbers as input. Check the alphabetic listings in the Reference Guide for specific requirements.

Name has no value
The variable *name* was used, but had not previously been assigned a value. Verify that the name used was correct (**shownames** prints the variables in memory). A variable is given a value by naming it on the input line of a procedure, and with the **make** and **name** commands.

Name is not true or false
Name was used as an input to a primitive requiring **true** or **false** as input (e.g. **if, ifelse**, **and**, **or** and **not).**

Name is the name of a primitive
A procedure was created using a name that LogoWriter already uses for a primitive (**forward**, **right**, etc.). Primitive names can't be used for your own procedures.

Name needs more inputs
Name was not given the inputs it requires. Check the primitive definitions to find what inputs *name* requires, or, if *name* is a procedure, check its definition on the flip side of the page.

No more room for text on the page
An instruction to paste or load caused the total number of characters on both sides of the page to exceed the maximum that the system will allow.

Not enough room to load
There is not enough room on the page to load the text file.

Number too big

On different computers, the maximum number that LogoWriter can use is different (they are all *very* big numbers).

Out of space

There is no free memory left. The free memory is related to, but is not the same as, the amount of space on the page, since variables and tools, for example, reside in a part of memory outside of the page. If this message occurs, leave the page (to save your procedures and graphics), and on a new, blank page, free some memory (**clearnames** removes variables from memory, and **cleartools** removes tool procedures).

Page is locked

The page (or file or directory) which you tried to erase or change is write-protected, and can't be erased or changed. **Unlock** removes write protection from a page.

Or, you tried to erase a directory which isn't empty. One idiosyncracy of the ProDOS operating system is that the same error message is sent when you try to erase a locked page or a non-empty directory. If an attempt to erase a directory results in this message, erase each page and file from the directory before you erase the directory itself.

Page name must be less than *number* characters

The name you used is too long for the computer to use as a file name. Choose a name which has fewer letters than the number in the message.

Pathname not found

There is no file or directory which matches the pathname used. Check that you typed the pathname correctly, or check the subdirectory and file names on the disk.

Please name this page

The page that is displayed needs a name before LogoWriter can put it in the scrapbook. If you want to keep the page, use **namepage** to name it; if not, use **cp** to clear the page before leaving it.

Stopped!!
The **stop** keys were pressed to halt an instruction or a procedure.

String too long
There are too many characters in a character string (a word). The maximum is approximately 32,000.

That's not a LogoWriter page
Getpage was given the name of a file that wasn't created by LogoWriter. Use **pagelist** to find the names of the LogoWriter pages on the disk.

There is a locked page with that name
The page on the screen is not locked, but the disk contains a locked page with the same name. To save the page, change the page name or swap disks.

There is already a file named *name*
The disk or directory already contains a file with that name. Use a different name, or erase the file with that name before trying to save.

There is already a page named *name*
Two pages cannot have the same name; choose a different name.

There is no file named *name*
The disk or directory contains no file with the name you used. Check that the spelling is corect, and that you used the correct filename extension (**.lwr** for LogoWriter pages, **.shp** for the Shapes page). LogoWriter's filing commands require the full filename as input.

There's no page named *name*
The name used as input to **getpage**, **load**, or **gettools** is not on the disk or in the current directory. To find out the names of the pages in your scrapbook, go to Contents or use **pagelist.**

Too few items in *name*

Name doesn't have the required number of elements. For instance, the fifth element of a list was requested (**item 5**), but the list contains only four items.

Too much inside ( ) 's

Parentheses were incorrectly used. For example, there are too many parentheses inside other parentheses. Simplify the instruction.

Wrong file type

The file name that you used as input is incompatible with the command. For example, you tried to use **loadpic** on a text file, when **loadpic** requires a graphic file.

Appendix 5
Glossary of LogoWriter and Computer Terms

ASCII
American Standard Code for Information Interchange. A code which represents characters numerically.

binary file
A file containing binary data (as compared to ASCII characters). Graphics files are often binary files, as are compiled program files.

brackets [ ]
Characters used to surround lists in LogoWriter.

buffer
An area of memory in which different types of information can be temporarily stored. The Clipboard is a buffer used to store text that is cut or copied. LogoWriter also has a shape buffer, used to store shapes that are cut or copied.

carriage return
A character which causes the cursor to move to the first position on the next line (ASCII character 13.)

command
A word that tells LogoWriter to do something. All LogoWriter instructions begin with a command.

command center
The bottom part of the screen where instructions are typed. It is not part of the page.

conditional
An instruction which causes LogoWriter to do one thing or another, depending on whether a condition is met. **If** and **ifelse** are used to build conditional expressions.

cursor
A small white square which indicates your place on the page or in the command center. LogoWriter has text cursors which show where the next letter typed will appear, and a Shapes page cursor which shows where the next shape-section will fill or empty.

directory
A file which holds information about other files on a disk, such as their name, location on the disk, size, and type.

down keys
The key combination which activates the cursor in the command center.

element
A component of a LogoWriter word or list. The elements of a list are words or other lists; the elements of a word are characters.

empty list
A list that has no elements. You write the empty list as [ ].

empty word
A word that has no characters. You write the empty word as "

file
An organized collection of information that has been stored on a disk. A LogoWriter page is a file that stores text and graphics together.

flip side
The side of a LogoWriter page on which you define procedures.

flip keys
The key combination which flips from the front side to the flip side of the page and back.

front side
The side of a LogoWriter page on which you do your drawing or writing. The display area of LogoWriter.

global variable
A variable which stays in memory even when no procedure is running and when pages are changed. In general, a variable is global when it is created with the **make** or **name** commands. A variable which remains in memory only while a procedure is running is a *local variable*. For more information, see the listings for **make** and **name.**

input(s)
The information that a LogoWriter primitive or procedure needs to run. It may be provided directly, by putting a word (numbers are words) or list after a primitive that needs an input, or indirectly, by using a reporter.

instruction
A LogoWriter statement that starts with a command; a command followed by its input(s), or by reporters which provide the inputs.

joystick
A game control device with a lever controlling two internal dials. One dial follows the x-axis movement of the lever, and the other follows the y-axis movement.

list
A sequence of words (numbers are words) or lists, or words and lists, that begins and ends with brackets. Lists are one of the basic LogoWriter units for handling information.

local variable
A variable whose value is in memory only while a procedure is running. A variable can be made local by naming it on the title line of a procedure. A variable whose value stays in memory even when no procedures are running is a *global variable*.

network
A set of interconnected computers which can share programs and information. Commonly, one main computer runs a program which determines the flow of information and prevents errors. All other programs, such as LogoWriter, run under this main program.

next screen keys
The key combination which causes the cursor on the page to move down exactly one screenful. On the Shapes page, the same keys cause the shapes to scroll one after the other.

object
A word (numbers are words) or a list. The units that can be accepted as inputs by LogoWriter primitives and procedures.

operation
Another term that is used to refer to reporters (primitives or procedures which report or output).

output
To give information to a preceding primitive or procedure in a LogoWriter instruction. A primitive or procedure which outputs is called a reporter.

paddle
A hand-held game control device with a single rotating dial and pushbutton.

page
A LogoWriter file that is part of your scrapbook. The front side of the page can contain pictures and text; the flip side can contain procedures. When you select a page, it is displayed on the top of the screen.

parentheses ( )
Characters which are used in LogoWriter instructions to group primitives or procedures in such a way as to control the sequence in which they are run. Parentheses can also be used with some LogoWriter primitives to extend the number of inputs they can accept.

parse
The process of converting a sequence of characters containing special characters into a list made of words.

pathname
A ProDOS term. The pathname is a complete name for a file, which incorporates the file's name and the name of the volume and subdirectories which contain the file. A file named **Sally**, in a volume named **Filedisk**, would have the pathname **%filedisk%sally**.

predicate
A term used for LogoWriter procedures which report **true** or **false**. They are also called "test reporters."

previous screen keys
The key combination that causes the cursor on the page to move up exactly one screenful. On the Shapes page, the same keys cause the shapes to scroll back one after the other.

primitive
A word that is built into LogoWriter.

procedure
A LogoWriter instruction or series of instructions which has a name.

quotation mark "
A special character which marks the beginning of a word. A quotation mark is essential if a word is to be used as an input to a LogoWriter primitive or procedure; otherwise, LogoWriter will try to interpret the word as a procedure to run.

recursive procedure
A procedure that runs itself as a subprocedure.

reporter
A LogoWriter primitive or procedure which can report (output) some information such as a number, word, or list.

scrapbook
A collection of LogoWriter pages on a single disk. A scrapbook always contains the Shapes page.

stop keys
The key combination which stops a LogoWriter procedure while it's running.

subdirectory
A directory which is within another directory, and which contains information on a certain group of disk files only.

subprocedure
A procedure which is run by another procedure; a procedure whose name is used in the definition of another procedure.

superprocedure
A procedure which uses one or more other procedures in its definition; a procedure that runs another procedure.

test reporter
A LogoWriter primitive or procedure which reports **true** or **false**, depending on the result of a test.

text file
A file containing only text (ASCII characters). Also called an ASCII file.

truncate
To cut off the last element(s) of a word or a number.

up keys
The key combination which activates the text cursor in the page.

variable
A name that stands for some value. A name with a colon (:) in front of it is a variable. There are two types of variables: global variables and local variables.

volume
A ProDOS term for a group of files on a disk. Simply put, a volume is a disk which has a name. ProDOS based programs use the volume name when referring to the disk.

word
A sequence of one or more characters, treated as a unit. Numbers are words in LogoWriter. Words are separated by spaces. A word can contain a special character such as a space only if the space is preceded by a backslash.

workspace
The part of the computer's memory that is available to hold variables and procedures as long as the computer is turned on.

write protected
A file or disk that has been set in such a way as to prevent any data from being changed or erased is write-protected. Information may be read from a write-protected file or disk, but may not be saved onto it. The protection may be contained within the file itself, or may be done physically, as when a tab is glued onto a disk.

Index

Special Characters

* asterisk, multiplication sign **2-5**
\ backslash **2-77, 3-25**
[ ] brackets **4-15, 4-21**
: colon **2-53**
/ division sign **2-5**
= equal sign **2-6**
> greater than sign **2-6**
< less than sign **2-6**
- minus sign **2-5**
( ) parentheses **4-19, 4-24**
+ plus sign **2-5**
" quotation mark **4-25**
| vertical bar **3-25**

A

addition **2-5**
all 2-7
and 2-7
Apple - combination keys **3-4, 3-40**
arctan 2-8
arrow keys **1-12, 1-19**
ascii 2-9
ASCII **4-21**
ask 2-9

B

back, bk 2-11
beginning of line keys **1-22**
bg 2-11
.bload 2-129
bottom 2-12
bottom of page keys **1-22, 1-24**
brackets [ ] **4-15, 4-21**
.bsave 2-130
buffer **4-21**
butfirst, bf 2-12
butlast, bl 2-13
button? 2-13
bye 2-111

C

.call 2-130
carriage return 2-44, 2-65, 4-21
cb 2-15
cc 2-15
cd 2-16
cf 2-16
cg 2-17
changing directories 2-119, 2-121
char 2-17
charunder 2-18
chdir 2-121
clean 2-18
clearevents 2-19
clearname 2-19
clearnames 2-20
cleartools 2-20
clipboard 2-21
colon : 2-53
color 2-21
colors, Apple *II* 3-5
colors, COMMODORE 64 3-50
colors, IBM PC
    CGA graphics 3-22
    EGA graphics 3-23
    PS/2 graphics 3-24
    version 1.1 3-45
colors, IBM PC*jr* 3-47
colorunder 2-22
command 4-21
command center 1-12
COMMODORE - combination keys 3-49
conditional 4-21
contents 2-22
Contents page 1-11, 3-10, 3-29
Contents page keys 1-24, 3-10, 3-29
control-key events 1-13, 2-103
copy 2-23
copy keys 1-21, 1-23
copyfile 2-111, 2-122
copying a disk 1-17
copying a page 1-15
copying a Shapes page 1-16
cos 2-23

cosine 2-23
count 2-24
cp 2-24
createdir 2-112
creating directories 2-112, 2-126
ct 2-25
ctrl - combination keys, IBM PC 3-20
ctrl - combination keys, IBM PC*jr* 3-47
cu 2-26
currentdir 2-123
cursor 4-22
cursorchar 2-26
cursorpos 2-27
cut 2-27
cut keys 1-21, 1-23

D

delete 2-29
.deposit 2-130, 2-131
directories 2-113, 2-123
disk 2-29, 3-7, 3-31
disk drives 1-16, 2-29, 2-81, 2-120, 3-7
distance 2-30, 2-89
division sign / 2-5, 3-13
dos 2-124
down keys 1-19
dspace 2-30

E

each 2-31
element 4-22
empty list 4-22
empty word 4-22
empty? 2-31
end 2-32
end key 3-21
end of line keys 1-22
eol 2-32
equal sign = 2-6
equal? 2-33
erase to end of line keys 1-21, 1-25
erasefile 2-113, 2-124

erasing a page **2**-33, **2**-113, **2**-124
erpage **2**-33
Escape key **1**-19, **1**-20, **1**-21, **1**-22
events **1**-13
.examine **2**-130, **2**-131

F

F1 key, COMMODORE 64 **1**-20
false **2**-35
fastturtle **2**-35
file **3**-6, **3**-26, **4**-22
filelist **2**-114,**2**-125, **3**-15, **3**-35
fill **2**-35
first **2**-36
flip **2**-36
flip keys **1**-19, **1**-23
flip side **1**-12, **4**-22
forward, fd **2**-36
found? **2**-37
fput **2**-37
front side **1**-12, **4**-22
front? **2**-37
Fn - combination keys, IBM PC*jr* **3**-46
Function keys, IBM PC **3**-20

G

game paddles **3**-5, **3**-21, **3**-51
garbage collection **2**-72
getpage **2**-39
getshapes **2**-39
gettools **2**-40
global variable **2**-53, **4**-22
graphics files **2**-115, **2**-125
greater than sign > **2**-6

H

heading **2**-41
help keys **1**-14, **1**-20
Help page **1**-14
home **2**-41
ht **2**-42

I

identical? 2-43
if 2-43
ifelse 2-44
.in 2-132
input 2-2, 4-23
insert 2-44
instruction 4-23
int 2-45
item 2-45

J

joystick 3-5, 3-21, 3-51

K

key? 2-47
keyboard, Apple *II* 3-4
Keyboard, Apple *II* + 3-40
keyboard, COMMODORE 64 3-27
keyboard, IBM PC 3-20
keyboard, IBM PC*jr* 3-46

L

label 2-49
label keys 1-20
last 2-49
Last Page 3-38
leavepage 2-50
left, lt 2-50
less than sign < 2-6
list 2-51
list 4-22
list? 2-51
load 2-51
loadpic 2-115, 2-125
loadtext 2-116, 2-128
local variable 4-23
lock 2-52
LogoWriter disk 1-11
lput 2-52

M

make 2-53
member? 2-54
minus 2-54
minus sign - 2-5
mkdir 2-126
multiplication sign * 2-5

N

name 2-55
name? 2-55
namepage, np 2-56
newpage 2-56
next screen keys 1-22, 1-23, 1-24
nextscreen 2-56
not 2-57
number? 2-57

O

object 4-24
online 2-116
open a line key 1-20
operation 4-24
or 2-59
.out 2-132
output, op 2-59
output 2-59, 4-24

P

paddle 2-61
page 1-12
    copying 1-15, 2-111, 2-122
    erasing 2-33, 2-113, 2-124
    saving 1-12, 2-77
pagelist 2-61
parentheses ( ) 4-24
parse 2-62
parse 2-62, 4-25
paste 2-63
paste keys 1-21, 1-23

pd 2-63
pe 2-64
PgDn keys 3-21
PgUp keys 3-21
plus sign + 2-5
pos 2-64
predicate 4-25
prefix 2-117
previous screen keys 1-22, 1-23, 1-24
prescreen 2-64
primitive 2-1
print, pr 2-65
printnames 2-65
printscreen 2-66
printtext 2-66
printtext80 2-67
procedure 4-24, 4-25
pu 2-67
px 2-68

Q

quitting LogoWriter 2-111, 2-124
quotation mark " 4-25

R

random 2-69
readchar 2-69
readlist ,rl 2-70
readlistcc, rlcc 2-71
recursive procedure 4-25
recycle 2-72
remainder 2-72
rename 2-118, 2-127
repeat 2-73
replace 2-73
reporter 4-25
restore 2-74
rg 2-75
rmdir 2-127
right, rt 2-75
round 2-75
run 2-75

S

savepage 2-77
savepic 2-118, 2-128
savetext 2-119, 2-128
saving a page **1**-12, **2**-77
scrapbook **1**-11
scrapbook disk **1**-15
  changing disks **1**-15, **2**-81, **2**-119, **3**-7
search 2-77
select 2-78
select keys **1**-21
selected 2-79
sentence, se 2-79
setbg 2-80
setc 2-80
setdisk 2-81, **3**-7
seth 2-81
setpos 2-82
setprefix 2-119, **3**-7
setsh 2-82
setslot 2-120
settc 2-82
settextpos 2-83
setx 2-83
sety 2-84
shade 2-84
shape 2-85
shapes **1**-14, **2**-85
Shapes page **1**-14
  copying a Shapes page **1**-16, **2**-111, **2**-121
Shapes page keys **1**-22
show 2-86
shownames 2-86
sin 2-87
sine 2-87
slot 2-120
slowturtle 2-87
sol 2-87
space 2-88
spacebar **1**-23
sqrt 2-89
sspace 2-90
st 2-90
stamp 2-91

startup 2-91, 4-7
Startup page **1**-15, **4**-7
stop 2-92
stop keys **1**-20, **1**-24
stopall 2-92
subprocedure 4-26
subtraction **2**-5, **2**-54
superprocedure 4-26

T

tab 2-93
Table of Frequencies **2**-96
tc 2-93
tell 2-93
text files **2**-116, **2**-126
test reporter 4-26
textlen 2-94
textpos 2-94
thing 2-95
to 2-95
tone 2-96
tone buffer **3**-43
tone? 2-97, **3**-43
toollist 2-97
tools **4**-9
tools pages 4-9
top 2-98
top of page keys **1**-22, **1**-24
towards 2-98
true 2-99
truncate 4-26
turtle-move keys **1**-20
tutorials **1**-11
type 2-99

U

undo 2-101
unlock 2-101
unselect 2-102
up keys **1**-19

V

variables **1**-13, **2**-53, **4**-26
.version **2**-132

W

wait **2**-103
when **2**-103
who **2**-104
word **2**-104
word **4**-27
word? **2**-105
workspace **1**-13, **2**-88

X

xcor **2**-107

Y

ycor **2**-109

3rd revision, August 1988